Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP2005/015815

International filing date: 30 August 2005 (30.08.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-256465
Filing date: 03 September 2004 (03.09.2004)

Date of receipt at the International Bureau: 13 October 2005 (13.10.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 Date of Application:	２００４年　９月　３日
出 願 番 号 Application Number:	特願２００４－２５６４６５
パリ条約による外国への出願に用いる優先権の主張の基礎となる出願の国コードと出願番号 The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is	ＪＰ２００４－２５６４６５
出 願 人 Applicant(s):	ソニー株式会社

２００５年　９月２８日

特許庁長官
Commissioner,
Japan Patent Office

中嶋 誠

【書類名】　特許願
【整理番号】　0490463303
【提出日】　平成16年 9月 3日
【あて先】　特許庁長官殿
【国際特許分類】　G09C 1/00
【発明者】
　【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
　【氏名】　白井　太三
【発明者】
　【住所又は居所】　ベルギー国、ケイ・ユー・ルーヴァン　リサーチアンドデベロップメント　フロート　ベゲインホフ　５８－５９　Ｂ－３０００　ルーヴァン　ルーベンカソリック大学
　【氏名】　バート　プレネール
【特許出願人】
　【識別番号】　000002185
　【氏名又は名称】　ソニー株式会社
【代理人】
　【識別番号】　100093241
　【弁理士】
　【氏名又は名称】　宮田　正昭
　【電話番号】　03-5541-7577
【選任した代理人】
　【識別番号】　100101801
　【弁理士】
　【氏名又は名称】　山田　英治
　【電話番号】　03-5541-7577
【選任した代理人】
　【識別番号】　100086531
　【弁理士】
　【氏名又は名称】　澤田　俊夫
　【電話番号】　03-5541-7577
【手数料の表示】
　【予納台帳番号】　048747
　【納付金額】　16,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9904833

【書類名】特許請求の範囲
【請求項１】
Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行する暗号処理装置であり、
非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行する構成を有し、
前記複数ラウンド各々に対応するＦ関数の線形変換部は、ｍ個の非線形変換部各々の出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行する構成であり、
少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々においては、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立であることを特徴とする暗号処理装置。
【請求項２】
前記暗号処理装置において、さらに、
前記逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が正方ＭＤＳ行列であることを特徴とする請求項１に記載の暗号処理装置。
【請求項３】
前記Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理のアルゴリズムは、ラウンド数２ｒの暗号処理アルゴリズムであり、
前記Ｆ関数の線形変換部は、
ｒ個の全ての偶数ラウンドおよびｒ個の全ての奇数ラウンドにおいて$2 \leqq q < r$のｑ種類の異なる正方ＭＤＳ行列を順次繰り返し適用した線形変換処理を実行する構成であることを特徴とする請求項１に記載の暗号処理装置。
【請求項４】
前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列であることを特徴とする請求項１に記載の暗号処理装置。
【請求項５】
前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列も正方ＭＤＳ行列となる正方ＭＤＳ行列であることを特徴とする請求項１に記載の暗号処理装置。
【請求項６】
前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する要素を全て含む正方ＭＤＳ行列Ｍから選択された行ベクトルによって構成される行列Ｍ'から抽出された列ベクトルによって構成される行列によって構成されていることを特徴とする請求項１に記載の暗号処理装置。
【請求項７】
Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行する暗号処理方法であり、
非線形変換処理および線形変換処理を実行するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行し、
前記複数ラウンド各々に対応するＦ関数の線形変換処理は、ｍ個の非線形変換部各々の出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行し、
少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々においては、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：La^{-1}，

Ｌｂ－１を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列による線形変換処理を実行することを特徴とする暗号処理方法。

【請求項８】

前記暗号処理方法において、さらに、

前記逆行列：Ｌａ－１，Ｌｂ－１を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が正方ＭＤＳ行列である正方ＭＤＳ行列による線形変換処理を実行することを特徴とする請求項７に記載の暗号処理方法。

【請求項９】

前記Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理のアルゴリズムは、ラウンド数２ｒの暗号処理アルゴリズムであり、

前記Ｆ関数の線形変換処理は、

ｒ個の全ての偶数ラウンドおよびｒ個の全ての奇数ラウンドにおいて２≦ｑ＜ｒのｑ種類の異なる正方ＭＤＳ行列を順次繰り返し適用した線形変換処理を実行することを特徴とする請求項７に記載の暗号処理方法。

【請求項１０】

前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列であることを特徴とする請求項７に記載の暗号処理方法。

【請求項１１】

前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列も正方ＭＤＳ行列となる正方ＭＤＳ行列であることを特徴とする請求項７に記載の暗号処理方法。

【請求項１２】

前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する要素を全て含む正方ＭＤＳ行列Ｍから選択された行ベクトルによって構成される行列Ｍ'から抽出された列ベクトルによって構成される行列によって構成されていることを特徴とする請求項７に記載の暗号処理方法。

【請求項１３】

Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行するコンピュータ・プログラムであり、

非線形変換処理および線形変換処理を実行するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するステップを有し、

前記複数ラウンド各々に対応するＦ関数の線形変換処理は、ｍ個の非線形変換部各々の出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行する線形変換ステップであり、

前記線形変換ステップにおいて、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々では、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：Ｌａ－１，Ｌｂ－１を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列による線形変換処理を実行することを特徴とするコンピュータ・プログラム。

【書類名】明細書
【発明の名称】暗号処理装置、および暗号処理方法、並びにコンピュータ・プログラム
【技術分野】
【０００１】
本発明は、暗号処理装置、および暗号処理方法、並びにコンピュータ・プログラムに関する。さらに詳細には、暗号解析処理、攻撃処理として知られる線形解析、差分解析に対する耐性を向上させた暗号処理装置、および暗号処理方法、並びにコンピュータ・プログラムに関する。
【背景技術】
【０００２】
昨今、ネットワーク通信、電子商取引の発展に伴い、通信におけるセキュリティ確保が重要な問題となっている。セキュリティ確保の１つの方法が暗号技術であり、現在、様々な暗号化手法を用いた通信が実際に行なわれている。
【０００３】
例えばＩＣカード等の小型の装置中に暗号処理モジュールを埋め込み、ＩＣカードと、データ読み取り書き込み装置としてのリーダライタとの間でデータ送受信を行ない、認証処理、あるいは送受信データの暗号化、復号を行なうシステムが実用化されている。
【０００４】
暗号処理アルゴリズムには様々なものがあるが、大きく分類すると、暗号化鍵と復号鍵を異なる鍵、例えば公開鍵と秘密鍵として設定する公開鍵暗号方式と、暗号化鍵と復号鍵を共通の鍵として設定する共通鍵暗号方式とに分類される。
【０００５】
共通鍵暗号方式にも様々なアルゴリズムがあるが、その１つに共通鍵をベースとして複数の鍵を生成して、生成した複数の鍵を用いてブロック単位（６４ビット，１２８ビットなど）のデータ変換処理を繰り返し実行する方式がある。このような鍵生成方式とデータ変換処理を適用したアルゴリズムの代表的なものが共通鍵ブロック暗号方式である。
【０００６】
代表的な共通鍵ブロック暗号のアルゴリズムとしては、例えば米国標準暗号としてのＤＥＳ（Data Encryption Standard）アルゴリズムがあり、様々な分野において広く用いられている。
【０００７】
ＤＥＳに代表される共通鍵ブロック暗号のアルゴリズムは、主として、入力データの変換を実行するラウンド関数部と、ラウンド関数（Ｆ関数）部の各ラウンドで適用する鍵を生成する鍵スケジュール部とに分けることができる。ラウンド関数部の各ラウンドで適用するラウンド鍵（副鍵）は、１つのマスター鍵（主鍵）に基づいて、鍵スケジュール部に入力されて生成され、各ラウンド関数部で適用される。
【０００８】
しかし、このような共通鍵暗号処理においては、暗号解析による鍵の漏洩が問題となっている。暗号解析または攻撃手法の代表的な手法として、ある差分を持つ入力データ（平文）とその出力データ（暗号文）を多数解析することにより各ラウンド関数における適用鍵を解析する差分解析（差分解読法または差分攻撃とも呼ばれる）や、平文と対応暗号文に基づく解析を行う線形解析（線形解読法または線形攻撃とも呼ばれる）が知られている。
【０００９】
暗号解析による鍵の解析が容易であるということは、その暗号処理の安全性が低いということになる。従来のＤＥＳアルゴリズムにおいては、ラウンド関数（Ｆ関数）部の線形変換部において適用する処理（変換行列）が、各段のラウンドにおいて等しいものであったため解析が行いやすく、結果として鍵の解析の容易性を招いているという問題がある。
【発明の開示】
【発明が解決しようとする課題】

【００１０】
本発明は、上記問題点に鑑みてなされたものであり、線形解析や差分解析に対する耐性の高い共通鍵ブロック暗号アルゴリズムを実現する暗号処理装置、および暗号処理方法、並びにコンピュータ・プログラムを提供することを目的とする。
【課題を解決するための手段】
【００１１】
本発明の第１の側面は、
Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行する暗号処理装置であり、
非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行する構成を有し、
前記複数ラウンド各々に対応するＦ関数の線形変換部は、ｍ個の非線形変換部各々の出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行する構成であり、
少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々においては、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立であることを特徴とする暗号処理装置にある。
【００１２】
さらに、本発明の暗号処理装置の一実施態様において、さらに、前記逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が正方ＭＤＳ行列であることを特徴とする。
【００１３】
さらに、本発明の暗号処理装置の一実施態様において、前記Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理のアルゴリズムは、ラウンド数２ｒの暗号処理アルゴリズムであり、前記Ｆ関数の線形変換部は、ｒ個の全ての偶数ラウンドおよびｒ個の全ての奇数ラウンドにおいて$2 \leqq q < r$のｑ種類の異なる正方ＭＤＳ行列を順次繰り返し適用した線形変換処理を実行する構成であることを特徴とする。
【００１４】
さらに、本発明の暗号処理装置の一実施態様において、前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列であることを特徴とする。
【００１５】
さらに、本発明の暗号処理装置の一実施態様において、前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列も正方ＭＤＳ行列となる正方ＭＤＳ行列であることを特徴とする。
【００１６】
さらに、本発明の暗号処理装置の一実施態様において、前記Ｆ関数の線形変換部において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する要素を全て含む正方ＭＤＳ行列Ｍから選択された行ベクトルによって構成される行列Ｍ'から抽出された列ベクトルによって構成される行列によって構成されていることを特徴とする。
【００１７】
さらに、本発明の第２の側面は、
Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行する暗号処理方法であり、
非線形変換処理および線形変換処理を実行するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行し、
前記複数ラウンド各々に対応するＦ関数の線形変換処理は、ｍ個の非線形変換部各々の

出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行し、

少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々においては、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列による線形変換処理を実行することを特徴とする暗号処理方法にある。

【００１８】

さらに、本発明の暗号処理方法の一実施態様において、さらに、前記逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が正方ＭＤＳ行列である正方ＭＤＳ行列による線形変換処理を実行することを特徴とする。

【００１９】

さらに、本発明の暗号処理方法の一実施態様において、前記Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理のアルゴリズムは、ラウンド数２ｒの暗号処理アルゴリズムであり、前記Ｆ関数の線形変換処理は、ｒ個の全ての偶数ラウンドおよびｒ個の全ての奇数ラウンドにおいて$2 \leqq q < r$のｑ種類の異なる正方ＭＤＳ行列を順次繰り返し適用した線形変換処理を実行することを特徴とする。

【００２０】

さらに、本発明の暗号処理方法の一実施態様において、前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列であることを特徴とする。

【００２１】

さらに、本発明の暗号処理方法の一実施態様において、前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列も正方ＭＤＳ行列となる正方ＭＤＳ行列であることを特徴とする。

【００２２】

さらに、本発明の暗号処理方法の一実施態様において、前記Ｆ関数の線形変換処理において適用する異なる複数の正方ＭＤＳ行列の各々は、該複数の正方ＭＤＳ行列を構成する要素を全て含む正方ＭＤＳ行列Ｍから選択された行ベクトルによって構成される行列Ｍ'から抽出された列ベクトルによって構成される行列によって構成されていることを特徴とする。

【００２３】

さらに、本発明の第３の側面は、

Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号処理を実行するコンピュータ・プログラムであり、

非線形変換処理および線形変換処理を実行するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するステップを有し、

前記複数ラウンド各々に対応するＦ関数の線形変換処理は、ｍ個の非線形変換部各々の出力するｎビット、総計ｍｎビットの入力に対する線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行する線形変換ステップであり、

前記線形変換ステップにおいて、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々では、異なる正方ＭＤＳ行列：Ｌａ，Ｌｂが適用され、かつ該正方ＭＤＳ行列の逆行列：La^{-1}，Lb^{-1}を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立である正方ＭＤＳ行列による線形変換処理を実行することを特徴とするコンピュータ・プログラムにある。

【００２４】
なお、本発明のコンピュータ・プログラムは、例えば、様々なプログラム・コードを実行可能なコンピュータ・システムに対して、コンピュータ可読な形式で提供する記憶媒体、通信媒体、例えば、ＣＤやＦＤ、ＭＯなどの記録媒体、あるいは、ネットワークなどの通信媒体によって提供可能なコンピュータ・プログラムである。このようなプログラムをコンピュータ可読な形式で提供することにより、コンピュータ・システム上でプログラムに応じた処理が実現される。
【００２５】
本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。
【発明の効果】
【００２６】
本発明の構成によれば、非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理において、複数ラウンド各々に対応するＦ関数の線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行するとともに、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列：Ｌａ，Ｌｂを適用し、かつ該正方ＭＤＳ行列の逆行列：Ｌａ$^{-1}$，Ｌｂ$^{-1}$を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立であるか、あるいは正方ＭＤＳ行列を構成する性質とした正方ＭＤＳ行列による線形変換処理を実行する構成としたので、共通鍵ブロック暗号における線形攻撃に対する耐性が向上し、暗号鍵等の解析困難性が高まることとなり、安全性の高い暗号処理が実現される。
【００２７】
さらに、本発明の構成によれば、非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理において、複数ラウンド各々に対応するＦ関数の線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行するとともに、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用する構成とし、これらの正方ＭＤＳ行列自身が、線形独立性を示すか、または正方ＭＤＳ行列を構成する構成としたので、アクティブＳボックスの寄与による同時差分キャンセルの発生しないことが保証され、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつである暗号化関数全体でのアクティブＳボックスの最少数を大きくすることが可能となる。本構成により、線形攻撃、差分攻撃の双方に対して耐性が向上し、より安全性の高い暗号処理が実現される。
【発明を実施するための最良の形態】
【００２８】
以下、本発明の暗号処理装置、および暗号処理方法、並びにコンピュータ・プログラムの詳細について説明する。なお、説明は、以下の項目順に行う。
１．共通鍵ブロック暗号アルゴリズムにおける差分解析処理
２．共通鍵ブロック暗号アルゴリズムにおける線形解析処理
３．本発明に基づく暗号処理アルゴリズム
（３－ａ）差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例
（３－ｂ）線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例
（３－ｃ）差分攻撃および線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例

【００２９】

［１．共通鍵ブロック暗号アルゴリズムにおける差分解析処理］

まず、ＤＥＳ（Data Encryption Standard）暗号処理に代表される共通鍵ブロック暗号アルゴリズムにおける差分解析処理の概要について、一般化した共通鍵ブロック暗号モデルを用いて説明する。

【００３０】

共通鍵ブロック暗号のアルゴリズムは、主として、入力データの変換を実行するラウンド関数部と、ラウンド関数部の各ラウンドで適用する鍵を生成する鍵スケジュール部とに分けることができる。ラウンド関数部の各ラウンドで適用する鍵（副鍵）は、１つのマスター鍵（主鍵）に基づいて、鍵スケジュール部に入力されて生成され、各ラウンド関数部で適用される。この共通鍵暗号方式の代表的な方式に米国連邦標準暗号方式としてのＤＥＳ（Data Encryption Standard）がある。

【００３１】

Ｆｅｉｓｔｅｌ構造と呼ばれる代表的な共通鍵ブロック暗号の構造について、図１を参照して説明する。

【００３２】

Ｆｅｉｓｔｅｌ構造は、変換関数の単純な繰り返しにより、平文を暗号文に変換する構造を持つ。平文の長さを２ｍｎビットとする。ただし、ｍ，ｎは共に整数である。初めに、２ｍｎビットの平文を、ｍｎビットの２つの入力データP_L（Plain-Left）１０１，P_R(Plain-Right)１０２に分割し、これを入力値とする。

【００３３】

Ｆｅｉｓｔｅｌ構造はラウンド関数とよばれる基本構造の繰り返しで表現され、各ラウンドに含まれるデータ変換関数はＦ関数１２０と呼ばれる。図１の構成では、Ｆ関数（ラウンド関数）１２０がｒ段繰り返された構成例を示している。

【００３４】

例えば第１番目のラウンドでは、ｍｎビットの入力データＸと、鍵生成部（図示せず）から入力されるｍｎビットのラウンド鍵$K_1$１０３がＦ関数１２０に入力され、Ｆ関数１２０におけるデータ変換処理の後にｍｎビットのデータＹを出力する。出力はもう片方の前段からの入力データ（第１段の場合は入力データP_L）と排他的論理和部１０４において、排他的論理和演算がなされ、ｍｎビットの演算結果が次のラウンド関数へと出力される。この処理、すなわちＦ関数を定められたラウンド数（ｒ）だけ繰り返し適用して暗号化処理が完了し、暗号文の分割データC_L（Cipher-Left）、C_R（Cipher-Right）が出力される。以上の構成より、Ｆｅｉｓｔｅｌ構造の復号処理はラウンド鍵を挿入する順序を逆にするだけでよく、逆関数を構成する必要がないことが導かれる。

【００３５】

各ラウンドの関数として設定されるＦ関数１２０の構成について、図２を参照して説明する。図２（ａ）は、１つのラウンドにおけるＦ関数１２０に対する入力および出力を示す図であり、図２（ｂ）は、Ｆ関数１２０の構成の詳細を示す図である。Ｆ関数１２０は、図２（ｂ）に示すように、非線形変換層と線形変換層を接続したいわゆるＳＰＮ型の構成を有する。

【００３６】

ＳＰＮ型のＦ関数１２０は、図２（ｂ）に示すように、非線形変換処理を実行する複数のＳボックス（Ｓ－ｂｏｘ）１２１を有する。ラウンド関数部の前段からのｍｎビットの入力値Ｘは、鍵スケジュール部から入力されるラウンド鍵K_iと排他的論理和が実行され、その出力がｎビットずつ非線形変換処理を実行する複数（ｍ個）のＳボックス１２１に入力される。各Ｓボックスでは、例えば変換テーブルを適用した非線形変換処理が実行される。

【００３７】

Ｓボックス１２１からの出力データであるｍｎビットの出力値Ｚは、線形変換処理を実行する線形変換部１２２に入力されて、例えばビット位置の入れ替え処理などの線形変換

処理が実行され、ｍｎビットの出力値Ｙを出力する。この出力値Ｙが前段からの入力データと排他的論理和され、次のラウンドのＦ関数の入力値とされる。

【００３８】

図２に示すＦ関数１２０は、入出力ビット長がｍ×ｎ（ｍ，ｎ：整数）ビットであり、非線形変換層はｎビットの入出力を持つ非線形変換層としてのＳボックス１２１は、ｍ個並列にならんだ構成を有し、線形変換層としての線形変換部１２２はｎ次の既約多項式で定義される２の拡大体ＧＦ（２^{n}）上の元を要素として持つｍ次の正方行列に基づく線形変換処理を実行する。

【００３９】

線形変換部１２２における線形変換処理に適用する正方行列の例を図３に示す。図３に示す正方行列１２５は、ｎ＝８，ｍ＝８の場合の例である。非線形変換部（Ｓボックス１２１）から出力されたｍ個のｎビットデータＺ［１］，Ｚ［２］，...，Ｚ［ｍ］に対してあらかじめ定められた正方行列１２５を適用した演算により線形変換が施され、Ｆ関数（ラウンド関数）出力としての、Ｙ［１］，Ｙ［２］，...，Ｙ［ｍ］が決定される。ただし、このとき各データの行列の要素に対する線形演算はあらかじめ定められた２の拡大体ＧＦ（２^{n}）上で行われる。

【００４０】

これまでのＦｅｉｓｔｅｌ型暗号では、すべての段のＦ関数に同じ線形変換層を用いているため、差分の伝播時に同時に複数の差分がキャンセルしてしまうという性質が存在した。背景技術の欄において説明したように、暗号解析手法の代表的な手法として、ある差分を持つ入力データ（平文）とその出力データ（暗号文）を多数解析することにより各ラウンド関数における適用鍵を解析する差分解析（あるいは差分解読法）が知られており、従来のＤＥＳ暗号アルゴリズム等の共通鍵ブロック暗号においては、Ｆ関数１２０部の線形変換部１２２において適用する処理（変換行列）を、各段のラウンドにおいて等しいものに設定しているため、差分解析が行いやすく、結果として鍵の解析の容易性を招いている。

【００４１】

差分の伝播時に、同時に複数の差分がキャンセルする例について、図４を参照して説明する。なお、本明細書においては、差分を表す場合にはΔ（デルタ）記号をつけて表す。

【００４２】

図４はｍ＝８，ｎ＝８の１２８ｂｉｔブロック暗号における３段の同時差分キャンセルの様子を説明する図である。ただし、図中では６４ｂｉｔのデータをバイト単位で区切ってベクトルとして表現し、それぞれの要素を１６進数で表記するものとする。

【００４３】

３段構成を持つＦ関数での同時差分キャンセルは、例えば以下のデータ状態１～４の設定メカニズムに基づいて発生する。以下に説明するメカニズムの発生するデータ状態は、多数の差分入力データを設定することで発生させることができるデータ状態であり、いわゆる差分解析における鍵（ラウンド鍵）の解析において発生し得る。

【００４４】

（状態１）

ｉラウンドへの入力差分の左半分は、すべてゼロである入力差分（ΔＸｉ－１＝（００，００，００，００，００，００，００，００））であり、右半分の入力差分がただひとつのＳ－ｂｏｘへの入力を除いてゼロである入力差分（ΔＸｉ＝（３４，００，００，００，００，００，００，００））であるとする。このデータ状態は、多数の差分入力データを設定することで、ｉラウンドにおいて、このようなデータ状態を得ることができるということである。

【００４５】

なお、ΔＸｉ＝（３４，００，００，００，００，００，００，００）の８つの各要素は、Ｆ関数中に構成されるｍ個（ｍ＝８）のＳボックス各々に対する入力差分に対応する。差分（３４）が第１Ｓボックス（図４中の（Ｓ１））に入力され、（００）が、第２～

８Ｓボックスに対する入力差分である。

【００４６】

なおゼロ（００）の入力差分を持つＳボックスの出力差分はゼロ（００）であり、差分データに関する限り、ゼロ（００）の入力差分を持つＳボックスは、何の作用も行なっていないものであり、アクティブでないすなわち非アクティブＳボックスと呼ばれる。一方、非ゼロの入力差分（図４の例では差分：３４）を持つＳボックスは、非ゼロの入力差分に対応した非線形変換結果を出力差分として発生させるので、アクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）と呼ばれる。

【００４７】

図４の例では、非ゼロの入力差分（３４）を入力する１つのアクティブＳボックス（Ｓ１）の出力差分（ｂ７）を発生させており、その他の非アクティブＳボックスＳ２～Ｓ８はゼロの入力差分（００）に基づいて出力差分（００）を発生させ、線形変換部の差分入力としている。

【００４８】

（状態２）

ｉラウンドへの非ゼロの入力差分（図４の例では差分：３４）を持つＳボックス（以下、アクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）と呼ぶ）からの出力差分は線形変換層で拡散されたのちＦ関数から出力（出力値＝ΔＹｉ）され、そのまま次のラウンドへの入力差分ΔＸｉ＋１となる。

【００４９】

図４の例における線形変換は、各ラウンドのＦ関数において共通する例えば図５に示すある特定の正方行列１２５による線形変換が実行されｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）を出力する。図５に示す線形変換構成から理解されるように、出力差分ΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）は、１つのアクティブＳボックス（Ｓ１）からの出力要素Ｚ［１］＝ｂ７にのみ依存した値として決定される。

【００５０】

このｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）は、図４に示す排他的論理和部１３１において、すべてゼロである入力差分（ΔＸｉ－１＝（００，００，００，００，００，００，００，００）と排他的論理和（ＸＯＲ）演算が実行され、演算結果が、次のラウンド（ｉ＋１）への入力差分ΔＸｉ＋１となる。

【００５１】

ｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、すべてゼロである入力差分ΔＸｉ－１＝（００，００，００，００，００，００，００，００）との排他的論理和（ＸＯＲ）結果は、ΔＹｉであるので、次のラウンド（ｉ＋１）への入力差分ΔＸｉ＋１＝ΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）となる。

【００５２】

（状態３）

ｉ＋１ラウンドのＦ関数からの出力差分ΔＹｉ＋１が、ｉラウンドでのＡｃｔｉｖｅ　Ｓ－ｂｏｘの位置にのみ非ゼロ値をもつ。このデータ状態は、多数の差分入力データを設定することで、このようなデータ状態を得ることができるということである。

【００５３】

すなわち、ΔＹｉ＋１＝（ａｄ，００，００，００，００，００，００，００）であり、ｉラウンドと同様、非ゼロの差分値（図４の例では差分：３４）を持つＳ－ｂｏｘの位置（第１Ｓボックス（Ｓ１））にのみ非ゼロ値をもつ。なお、明らかにａｄ≠００である。

【００５４】

（状態４）

ｉ＋２ラウンドのアクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）（Ｓ１）の出力差分がｉラウンドでのアクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）（Ｓ１）の出力差分と一致した場合、すなわち、図４に示すようにｉ＋２ラウンドのアクティブＳボックス（（Ｓ１）の出力差分がｂ７となり、ｉラウンドでのアクティブＳボックス（Ｓ１）の出力差分（ｂ７）と一致する。このデータ状態は、多数の差分入力データを設定することで、このようなデータ状態を得ることができるということである。

【００５５】

このデータ状態が発生すると、ｉ＋２ラウンドのＦ関数の出力差分ΔＹｉ＋２＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）が、２つ前のラウンドであるｉラウンドのＦ関数の出力差分ΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と一致することになる。

【００５６】

この結果、排他的論理和部１３３では、

ΔＸｉ＋１＝ΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、

ΔＹｉ＋２＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、

の同一の値同士の排他的論理和演算が実行されることになり、排他的論理和演算結果としてオール０の値を出力する。

【００５７】

その結果、次の段（ラウンドｉ＋３）への出力差分の前段（ｉ＋２ラウンド）からの左の入力差分ΔＸｉ＋３＝（００，００，００，００，００，００，００，００）となる。

【００５８】

このラウンドｉ＋３への左入力ΔＸｉ＋３＝（００，００，００，００，００，００，００，００）は、ラウンドｉへの左入力ΔＸｉ－１＝（００，００，００，００，００，００，００，００）と同様オールゼロであり、ラウンドｉ＋３以降のラウンドにおいても、ラウンドｉ～ｉ＋２と同様の処理が繰り返される可能性がある。

【００５９】

この結果、ラウンド数の伸びに対してアクティブＳボックスの数が増大せず、差分攻撃に対する強度がそれほど伸びないという問題を発生させる。

【００６０】

共通鍵ブロック暗号において、差分攻撃に対する強度指標のひとつとして、暗号化関数全体でのアクティブＳボックスの最少数が知られている。アクティブＳボックス数の最少数が大きいほど差分攻撃に対する耐性が高いと判断される。

【００６１】

前述したように、差分解析（差分攻撃）においては、ある差分を持つ入力データ（平文）とその出力データ（暗号文）を多数設定してこの対応を解析することにより各ラウンド関数における適用鍵を解析する手法であり、この差分解析において、アクティブＳボックスの数を少なくできれば、解析が容易となり、解析プロセス数を削減できる。

【００６２】

上述の図４を参照した例では、第１のＳボックス（Ｓ１）のみがアクティブＳボックスであるパターンの発生状態を提示したが、その他のＳボックス（Ｓ２～Ｓ８）についても、差分解析の入力データの設定によって、各ＳボックスのみをアクティブＳボックスとした設定が可能であり、このような差分解析プロセスを実行することにより、各Ｓボックスの非線形変換処理の解析、さらにＦ関数に対して入力されるラウンド鍵の解析が可能となる。

【００６３】

このような差分解析に対する耐性を向上させるためには、アクティブＳボックスの数が常に多い状態を維持すること、すなわち、アクティブＳボックスの最少数が多いことが必要である。

【００６４】

図４を参照して説明した例において、右から左へ入力を行なうＦ関数、すなわち、第ｉ

ラウンドと第ｉ＋２ラウンドのみをアクティブＳボックス算出処理対象ラウンドとしてみた場合、アクティブＳボックス数はわずか２であり、左から右へ入力を行なうＦ関数、すなわち、第ｉ＋1ラウンドではアクティブＳボックス数が８であるものの、同時差分キャンセルにより第ｉ＋３ラウンドでのアクティブＳボックス数が０となってしまうため、差分解析による各Ｓボックスの非線形変換処理の解析処理が容易となる。

【００６５】

図４に示す共通鍵ブロック暗号アルゴリズムは、各ラウンドにおける線形変換部において適用する線形変換行列が等しいものであり、この構成に起因して、特に右から左へ入力を行うＦ関数におけるわずか２つのアクティブＳボックスにより同時差分キャンセルの発生可能性を引き起こしている。従って、ラウンド数の伸びに対してアクティブＳボックスの最少数が十分に増大せず、差分攻撃に対する強度がそれほど伸びないという問題がある。

【００６６】

次に、同様に、同じ線形変換行列をすべての段（ラウンド）のＦ関数に用いる構成において、５ラウンドにまたがる同時差分キャンセルの発生メカニズムについて、図６を参照して説明する。

【００６７】

図６はｍ＝８，ｎ＝８の１２８ｂｉｔブロック暗号における５段の同時差分キャンセルの様子を説明する図である。ただし、図中では６４ｂｉｔのデータをバイト単位で区切ってベクトルとして表現し、それぞれの要素を１６進数で表記するものとする。

【００６８】

５段構成を持つＦ関数での同時差分キャンセルは、例えば以下のデータ状態１～７の設定メカニズムに基づいて発生する。以下に説明するメカニズムの発生するデータ状態は、多数の差分入力データを設定することで発生させることができるデータ状態であり、いわゆる差分解析における鍵（ラウンド鍵）の解析において発生し得る。

【００６９】

（状態１）

ｉラウンドへの入力差分の左半分は、すべてゼロである入力差分（ΔＸｉ－１＝（００，００，００，００，００，００，００，００））であり、右半分の入力差分がただひとつのＳ－ｂｏｘへの入力を除いてゼロである入力差分（ΔＸｉ＝（３４，００，００，００，００，００，００，００））であるとする。このデータ状態は、多数の差分入力データを設定することで、ｉラウンドにおいて、このようなデータ状態を得ることができるということである。

【００７０】

なお、ΔＸｉ＝（３４，００，００，００，００，００，００，００）の８つの各要素は、Ｆ関数中に構成されるｍ個（ｍ＝８）のＳボックス各々に対する入力差分に対応する。（３４）が第１Ｓボックス（図６中の（Ｓ１））に入力され、（００）が、第２～８Ｓボックスに対する入力差分である。

【００７１】

なお前述したように、ゼロ（００）の入力差分を持つＳボックスの出力差分はゼロ（００）であり、差分データに関する限り、ゼロ（００）の入力差分を持つＳボックスは、何の作用も行なっていないものであり、アクティブでないすなわち非アクティブＳボックスと呼ばれる。一方、非ゼロの入力差分（図６の例では差分：３４）を持つＳボックス（Ｓ１）のみが、非ゼロの入力差分に対応した非線形変換結果を出力差分として発生させるので、アクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）である。

【００７２】

図６の例では、非ゼロの入力差分（３４）を入力する１つのアクティブＳボックス（Ｓ１）の出力差分（ｂ７）を発生させており、その他の非アクティブＳボックスＳ２～Ｓ８はゼロの入力差分（００）に基づいて出力差分（００）を発生させ、線形変換部の差分入力としている。

【００７３】
（状態２）
ｉラウンドへの非ゼロの入力差分（図４の例では差分：３４）を持つＳボックス（以下、アクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）と呼ぶ）からの出力差分は線形変換層で拡散されたのちＦ関数から出力（出力値＝ΔＹｉ）され、そのまま次のラウンドへの入力差分ΔＸｉ＋１となる。
【００７４】
図６の例において、各ラウンドに共通の例えば図５に示すある特定の正方行列１２５による線形変換が実行されｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）を出力する。
【００７５】
ｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）は、図６に示す排他的論理和部１４１において、すべてゼロである入力差分（ΔＸｉ－１＝（００，００，００，００，００，００，００，００）と排他的論理和（ＸＯＲ）演算が実行され、演算結果が、次のラウンド（ｉ＋１）への入力差分ΔＸｉ＋１となる。
【００７６】
ｉラウンドのＦ関数出力差分としてのΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、すべてゼロである入力差分（ΔＸｉ－１＝（００，００，００，００，００，００，００，００）との排他的論理和（ＸＯＲ）結果は、ΔＹｉであるので、次のラウンド（ｉ＋１）への入力差分ΔＸｉ＋１＝ΔＹｉ＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）となる。
【００７７】
（状態３）
ｉ＋１ラウンドのＦ関数からの出力差分ΔＹｉ＋１が、ｉラウンドでのＡｃｔｉｖｅ　Ｓ－ｂｏｘの位置にのみ非ゼロ値をもつ。このデータ状態は、多数の差分入力データを設定することで、このようなデータ状態を得ることができるということである。
【００７８】
すなわち、ΔＹｉ＋１＝（３４，００，００，００，００，００，００，００）であり、ｉラウンドと同様、非ゼロの差分値（図６の例では差分：３４）を持つＳ－ｂｏｘの位置（第１Ｓボックス（Ｓ１））にのみ非ゼロ値をもつ。
【００７９】
（状態４）
ｉ＋２ラウンドのＦ関数に対する入力は、ΔＸｉ＝（３４，００，００，００，００，００，００，００）と、ΔＹｉ＋１＝（３４，００，００，００，００，００，００，００）との排他的論理和部１４２における排他的論理和結果、すなわち、同一データ同士の排他的論理和であり、オールゼロの入力、ΔＸｉ＋２＝（００，００，００，００，００，００，００，００）となり、その結果、ｉ＋２ラウンドのＦ関数からの出力差分も、オールゼロの出力差分、ΔＹｉ＋２＝（００，００，００，００，００，００，００，００）となる。
【００８０】
（状態５）
ｉ＋３ラウンドのＦ関数に対する入力は、ΔＸｉ＋１＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、オールゼロのｉ＋２ラウンドのＦ関数出力差分ΔＹｉ＋２＝（００，００，００，００，００，００，００，００）との排他的論理和部１４３における排他的論理和結果であり、ｉ＋３ラウンドのＦ関数に対する入力ΔＸｉ＋３＝ΔＸｉ＋１＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）となる。
【００８１】
（状態６）
ｉ＋３ラウンドのＦ関数出力差分が、ΔＹｉ＋３＝（４３，００，００，００，００，

００，００，００）となり、オールゼロのΔＸｉ＋２＝（００，００，００，００，００，００，００，００）との排他的論理和部１４４における排他的論理和の結果としてのΔＸｉ＋４＝ΔＹｉ＋３＝（４３，００，００，００，００，００，００，００）がｉ＋４ラウンドのＦ関数入力差分となる。

【００８２】

（状態７）

ｉ＋４ラウンドのアクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）（Ｓ１）の出力差分がｉラウンドでのアクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）（Ｓ１）の出力差分と一致した場合、すなわち、図６に示すようにｉ＋４ラウンドのアクティブＳボックス（（Ｓ１）の出力差分がｂ７となり、ｉラウンドでのアクティブＳボックス（Ｓ１）の出力差分（ｂ７）と一致する。このデータ状態は、多数の差分入力データを設定することで、このようなデータ状態を得ることができるということである。

【００８３】

このデータ状態が発生すると、ｉ＋４ラウンドのＦ関数の出力差分ΔＹｉ＋４＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）が、２つ前のラウンドであるｉ＋２ラウンドの排他的論理和部１４３の出力差分ΔＸｉ＋３＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と一致することになる。

【００８４】

この結果、排他的論理和部１４５では、

ΔＸｉ＋３＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、

ΔＹｉ＋４＝（９８，ｃ４，ｂ４，ｄ３，ａｃ，７２，０ｆ，３２）と、

の同一の値同士の排他的論理和演算が実行されることになり、排他的論理和演算結果としてオール０の値を出力する。

【００８５】

その結果、次の段（ラウンドｉ＋５）への入力差分は、ΔＸｉ＋５＝（００，００，００，００，００，００，００，００）として設定される。

【００８６】

このラウンドｉ＋５への左入力ΔＸｉ＋５＝（００，００，００，００，００，００，００，００）は、ラウンドｉへの左入力ΔＸｉ－１＝（００，００，００，００，００，００，００，００）と同様オールゼロであり、ラウンドｉ＋５以降のラウンドにおいても、ラウンドｉ～ｉ＋４と同様の処理が繰り返される可能性がある。

【００８７】

この結果、ラウンド数の伸びに対してアクティブＳボックスの数が増大せず、差分攻撃に対する強度がそれほど伸びないという問題を発生させる。

【００８８】

前述したように、差分解析（差分攻撃）においては、ある差分を持つ入力データ（平文）とその出力データ（暗号文）を多数設定してこの対応を解析することにより各ラウンド関数における適用鍵を解析する手法であり、この差分解析において、アクティブＳボックスの数を少なくできれば、解析が容易となり、解析プロセス数を削減できる。

【００８９】

上述の図６を参照した例において、右から左へ入力を行なうＦ関数、すなわち、第ｉラウンドと第ｉ＋２ラウンド、第ｉ＋４ラウンドのみをアクティブＳボックス算出処理対象ラウンドとしてみた場合、アクティブＳボックス数は、第ｉラウンド＝１、第ｉ＋２ラウンド＝０、第ｉ＋４ラウンド＝１の合計わずか２であり、左から右へ入力を行なうＦ関数、すなわち第ｉ＋1ラウンドおよび第ｉ＋３ラウンドではアクティブＳボックス数が８であるものの、同時差分キャンセルにより第ｉ＋５ラウンドでのアクティブＳボックス数が０となってしまうため、差分解析による各Ｓボックスの非線形変換処理の解析、および、Ｆ関数に対する入力ラウンド鍵の解析処理が比較的、容易となる。

【００９０】

図６を参照した例では、第１のＳボックス（Ｓ１）のみがアクティブＳボックスである

パターンの発生状態を提示したが、その他のＳボックス（Ｓ２～Ｓ８）についても、差分解析の入力データの設定によって、各ＳボックスのみをアクティブＳボックスとした設定が可能であり、このような差分解析プロセスを実行することにより、各Ｓボックスの非線形変換処理の解析、さらにＦ関数に対して入力されるラウンド鍵の解析が可能となる。

【００９１】

図４および図６を参照して、３および５ラウンドの場合の同時差分キャンセルの発生例を説明したが、任意のラウンド数に一般化して同時差分キャンセルを定義すると以下のように定義することができる。図７を参照して、任意のラウンド数における同時差分キャンセルの定義について説明する。なお、図７は、フェイステル（Ｆｅｉｓｔｅｌ）構造の共通鍵ブロック暗号を実行するフェイステル（Ｆｅｉｓｔｅｌ）構造の１つおきのラウンド（ｉ，ｉ＋２，ｉ＋４，・・・，ｉ＋２ｊ）を示している。

【００９２】

「定義」

フェイステル（Ｆｅｉｓｔｅｌ）構造のラウンドｉでの入力差分の半分（ＰＬまたはＰＲ）が０（図７において、ΔＸｉ＝（００，００，００，００，００，００，００，００））であり、そこにｉ＋２ｊラウンド（ｊ＝０，１，２，...）のＦ関数の出力差分が排他的論理和部で演算されていく過程において、あるラウンドｉ＋２ｋにおいて、排他的論理和の結果が０（図７において、ΔＸｉ＋２ｊ＋１＝（００，００，００，００，００，００，００，００））になった場合を”同時差分キャンセル”と呼ぶ。

【００９３】

その時、ｉ，ｉ＋２，ｉ＋４，．．，ｉ＋２ｋラウンドのＦ関数の中に存在するアクティブＳボックス（Ａｃｔｉｖｅ　Ｓ－ｂｏｘ）のことを”同時差分キャンセルを発生させたアクティブＳボックス”と呼ぶものとし、ベクトルＡの非ゼロの要素数をハミングウェイトｈｗ（Ａ）と定義すると、同時差分キャンセルを発生させるアクティブＳボックスの数ａは、以下の式として表せる。

【数１】

$$a=\sum_{j=0}^{k} hw(\Delta X_{i+2j})$$

【００９４】

前述の３ラウンド、５ラウンドでの例ではともに同時差分キャンセルを発生させたアクティブＳボックス数は２、すなわちａ＝２である。

【００９５】

前述したように、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつが暗号化関数全体でのアクティブＳボックスの最少数であり、アクティブＳボックス数の最少数が大きいほど差分攻撃に対する耐性が高いと判断される。

【００９６】

しかし、ＤＥＳアルゴリズムのように同じ線形変換行列をすべての段のＦ関数に用いる

構成では、図４、図６を参照して説明したように、わずか２つのアクティブＳボックスにより同時差分キャンセルが発生してしまう可能性があった。そのような性質があるためラウンド数の伸びに対してアクティブＳボックスの最少数が十分に増大せず、差分攻撃に対する強度がそれほど伸びないという問題があった。

【００９７】

［２．共通鍵ブロック暗号アルゴリズムにおける線形解析処理］

差分解析処理は、上述したように、解析の実行者が一定の差分を持つ入力データ（平文）を容易し、その対応する出力データ（暗号文）を解析することが必要となる。線形解析処理は、一定の差分を持つ入力データ（平文）を準備する必要はなく、所定量以上の入力データ（平文）と対応する出力データ（暗号文）とに基づいて解析を行う。

【００９８】

前述したように、共通鍵ブロック暗号アルゴリズムでは非線形変換部としてのＳボックスを有し、入力データ（平文）と対応する出力データ（暗号文）との線形関係はないが、線形解析では、このＳボックスの入出力を線形近似し、多数の入力データ（平文）と対応する出力データ（暗号文）の構成ビット値の線形関係を解析し、候補となる鍵を絞り込むことによって解析を行う。線形解析においては、特定の差分を持つ入力データを準備することが必要ではなく、多数の平文と対応暗号文を容易することで、解析が可能となる。

【００９９】

［３．本発明に基づく暗号処理アルゴリズム］

以下、本発明の暗号処理アルゴリズムについて説明する。本発明の暗号処理アルゴリズムは、上述した線形解析、差分解析等の攻撃に対する耐性を向上させた構成、すなわち、鍵解析の困難性を高め、安全性を向上させた構成を持つ。

【０１００】

本発明に係る暗号処理アルゴリズムの１つの特徴は、従来のＤＥＳアルゴリズムの如く各ラウンドのＦ関数に構成される線形変換部に共通の処理（変換行列）を適用した構成とせず、複数の異なる正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を設定した構成としたことである。具体的には、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用した線形変換処理を実行する構成を持つ。

【０１０１】

本発明に係る暗号処理アルゴリズムは、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列の性質を利用し、少ないアクティブＳボックスに基づく同時差分キャンセルが起こらない、または起こりにくい構造を実現し、アクティブＳボックスの最小数を増大させ、差分攻撃に対してより強い共通鍵ブロック暗号処理を実現する。あるいは、既知平文攻撃として行なわれる線形解析についての困難性も高めた構成を持つ。

【０１０２】

本発明の暗号処理アルゴリズムは、図１、２を参照して説明したＳＰＮ型のＦ関数を有するＦｅｉｓｔｅｌ構造と呼ばれる代表的な共通鍵ブロック暗号の構造、すなわち、非線形変換部および線形変換部を有するＳＰＮ型のＦ関数の複数ラウンドに渡る単純な繰り返しにより、平文を暗号文に変換する、あるいは暗号文を平文変換する構造を適用している。

【０１０３】

例えば、平文の長さを２ｍｎビット（ただし、ｍ，ｎは共に整数）として、２ｍｎビットの平文を、ｍｎビットの２つのデータＰＬ（Plain-Left），ＰＲ(Plain-Right)に分割し、これを入力値として、各ラウンドにおいて、Ｆ関数を実行させるものであり、Ｆ関数は、図２を参照して説明したように、Ｓボックスからなる非線形変換部と、線形変換部を接続したＳＰＮ型を持つＦ関数である。

【０１０４】

本発明の構成においては、Ｆ関数中の線形変換部において適用する線形変換処理のため

の行列として、複数の異なる正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列から選択された行列を各ラウンドのＦ関数の線形変換部において適用する行列として設定する。具体的には、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用する。

【０１０５】

正方ＭＤＳ行列について説明する。正方ＭＤＳ行列とは以下の（ａ），（ｂ）の性質を満たす行列をいう。

（ａ）正方行列である

（ｂ）行列に含まれるすべての部分行列（ｓｕｂｍａｔｒｉｘ）の行列式（ｄｅｔｅｒｍｉｎａｎｔ）が０でない、すなわち、ｄｅｔ（ｓｕｂｍａｔｒｉｘ）≠０

【０１０６】

上記（ａ），（ｂ）の条件を満足する行列を正方ＭＤＳ行列と呼ぶ。

共通鍵ブロック暗号の各ラウンドで実行するＦ関数に対する入出力ビット長がｍ×ｎ（ｍ，ｎ：整数）ビットであり、Ｆ関数内に構成される非線形変換部がｎビットの入出力を持つｍ個のＳボックスにより構成され、線形変換部がｎ次の既約多項式で定義される２の拡大体ＧＦ（2^n）上の元を要素として持つｍ次の正方行列に基づく線形変換処理を実行する場合の、正方ＭＤＳ行列の一例を図８に示す。図８に示す正方ＭＤＳ行列の例は、ｎ＝８，ｍ＝８の正方ＭＤＳ行列の例である。

【０１０７】

上記（ａ），（ｂ）を満足する正方ＭＤＳ行列は、ベクトルＡの非ゼロの要素数をハミングウェイトｈｗ（Ａ）とし、Ｍをｍ次の正方ＭＤＳ行列とし、ｘを正方ＭＤＳ行列Ｍへの入力ベクトルとした場合、以下の不等式（式１）を満たすことになる。

ｈｗ（ｘ）＋ｈｗ（Ｍｘ）≧ｍ＋１……（式１）

【０１０８】

上記式（式１）は、正方ＭＤＳ行列（Ｍ）によって線形変換される入力データｘの非ゼロの要素数ｈｗ（ｘ）と、正方ＭＤＳ行列（Ｍ）によって線形変換された出力データＭｘの非ゼロの要素数ｈｗ（Ｍｘ）の総数が、正方ＭＤＳ行列の次数ｍより大となるということを意味している。

【０１０９】

なお、正方ＭＤＳ行列という名は正方ＭＤＳ－ｃｏｄｅ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ　Ｃｏｄｅ）の生成行列の標準形の右半分が上記条件を満足していることから名づけているものである。

【０１１０】

１つの行列をすべてのＦ関数に組み込むという従来の構成でも線形変換行列に正方ＭＤＳ行列を用いることで、正方ＭＤＳ行列でない行列を用いる場合に比べてアクティブＳボックス数の最少数を比較的高水準に保持することができるということは知られている。

【０１１１】

本発明では、各ラウンドのＦ関数には正方ＭＤＳ行列の条件を満たす行列を利用し、さらにラウンドごとに異なる行列を設定する方法を提案する。具体的には、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用する。

【０１１２】

以下に、段数（ラウンド数）が２ｒ（ｒは整数）のＦｅｉｓｔｅｌ型共通鍵ブロック暗号において、差分攻撃に対する耐性をより高めた複数の構成例について説明する。

【０１１３】

なお、以下の説明において、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成のｊ段目のＦ関数における線形変換部で適用する線形変換行列をＭＬＴｊとして表すものとする。

【０１１４】

本発明の構成では、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号

処理構成における各段のＦ関数中の線形変換部において適用する線形変換処理のための行列として、複数の異なる正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列から選択された行列を各ラウンドのＦ関数の線形変換部において適用する行列として設定する。具体的には、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用する。

【０１１５】

具体的には、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成に対応して、ｒ以下のｑ個の正方ＭＤＳ行列：Ｌ１，Ｌ２，・・，Ｌｑを生成し、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成における奇数段目のＦ関数中の線形変換部において適用する線形変換処理のための行列として、上位段のＦ関数から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。さらに、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。

【０１１６】

本設定を適用した構成例を図９に示す。図９は、段数（ラウンド数）が２ｒ＝１２、すなわちｒ＝６のＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成とした場合、ｑ＝３、すなわち、１２段のラウンド数を持つＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成において３種類の異なる正方ＭＤＳ行列を配置した構成例として、各ラウンドのＦ関数部の線形変換部に設定する正方ＭＤＳ行列（Ｌ１，Ｌ２，Ｌ３）を示している。

【０１１７】

図９の構成は、２ｍｎビットの平文を、ｍｎビットの２つのデータＰＬ（Plain-Left），ＰＲ(Plain-Right)に分割し、これを入力値として、各ラウンドにおいて、Ｆ関数を実行させる構成であり、第１ラウンドのＦ関数４０１およびその他のラウンドのＦ関数も、すべて図２を参照して説明したように、Ｓボックスからなる非線形変換部と、線形変換部を接続したＳＰＮ型を持つＦ関数である。

【０１１８】

図９の設定例はｒ＝６，ｑ＝３であり、各Ｆ関数内に示す記号Ｌｎは正方ＭＤＳ行列４０２を示している。すなわち、Ｌ１，Ｌ２，Ｌ３は、それぞれ異なる３種類の正方ＭＤＳ行列を示し、各Ｆ関数の線形変部において線形変換処理に適用する正方ＭＤＳ行列を示している。

【０１１９】

線形変換行列ＭＬＴｊの設定処理シーケンスについて、図１０を参照して説明する。

【０１２０】

［ステップＳ２１］

ラウンド数２ｒの半数ｒに対してｒ以下の数ｑ、すなわち、

ｑ≦ｒとなる数ｑを選択する。ただし、ｑは２以上の整数である。

［ステップＳ２２］

ｑ個のＧＦ（2^n）上のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを生成する。ｑ個のＧＦ（2^n）上のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑの生成処理手法についての詳細は、後段で説明する。

【０１２１】

ステップＳ２２において、ｑ個のＧＦ（2^n）上のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成した後、次に、以下の正方ＭＤＳ行列設定処理を実行する。

［ステップＳ２３］

２ｉ－１（１≦ｉ≦ｒ）段目の線形変換行列MLT_{2i-1}にＬ（ｉ－１ｍｏｄｑ）＋１を設定する。

［ステップＳ２４］

２ｉ（１≦ｉ≦ｒ）段目の線形変換行列にMLT_{2i}に$MLT_{2r-2i+1}$を設定する。

【０１２２】
例えば、図９に示す構成例、すなわち、１２段（ｒ＝６）でありｑ＝３とした場合は、
ＭＬＴ１＝Ｌ１，　ＭＬＴ２＝Ｌ３
ＭＬＴ３＝Ｌ２，　ＭＬＴ４＝Ｌ２
ＭＬＴ５＝Ｌ３，　ＭＬＴ６＝Ｌ１
ＭＬＴ７＝Ｌ１，　ＭＬＴ８＝Ｌ３
ＭＬＴ９＝Ｌ２，　ＭＬＴ１０＝Ｌ２
ＭＬＴ１１＝Ｌ３，　ＭＬＴ１２＝Ｌ１
の設定となる。
【０１２３】
このように、本発明の暗号処理装置においては、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成に対応して、ｒ以下のｑ個の正方ＭＤＳ行列：Ｌ１，Ｌ２，・・，Ｌｑを生成し、奇数段目については上位段のＦ関数から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する構成としている。
【０１２４】
次に、図１０の処理フローにおけるステップＳ２２のｑ個のＧＦ（2^n）上のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑの生成処理およびＦ関数への設定構成の詳細について説明する。なお、説明は以下の項目に沿って行なう。
（３－ａ）差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例
（３－ｂ）線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例
（３－ｃ）差分攻撃および線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例
【０１２５】
［（３－ａ）差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例］
まず、差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例として、３つの処理例ａ１，ａ２，ａ３について説明する。
【０１２６】
（処理例ａ１）
差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例の第１の例について説明する。まず、図１１に示すフローチャートを参照して正方ＭＤＳ行列の生成処理について説明する。
【０１２７】
［ステップＳ１０１］
入力：必要な正方ＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、
ＧＦ（2^n）上で、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑをランダムに生成する。なお、図１１に示すフローでは、ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。
【０１２８】
［ステップＳ１０２］
ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑに含まれるｑｍ個の列の任意のｍ個を取り出したときに、線形独立になっているかどうかをチェックする。チェックに通過したらステップＳ１０３に進む、そうでない場合はステップＳ１０１にもどる。
［ステップＳ１０３］
ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。

【０１２９】
以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。
【０１３０】
このようにして生成したｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定する。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。
【０１３１】
このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。
【０１３２】
本構成により、
（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、
（ｂ）暗号化関数内の奇数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが独立であること、
（ｃ）偶数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが独立であること、
これら（ａ）～（ｃ）が保証されるため、複数段のラウンド数を持つＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成において、連続する２ｑ－１ラウンドにおいて、ｍ個以下のアクティブＳボックスの寄与による同時差分キャンセルは発生しないことが保証される。よって暗号化関数全体のアクティブＳボックス数の最小値が増大する。
【０１３３】
このように、本処理例によって、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつである暗号化関数全体でのアクティブＳボックスの最少数を大きくすることが可能となり、結果として、差分解析（差分攻撃）を行なった場合のアクティブＳボックスの数が増大し、解析の困難性が高まることになる。従って鍵の解析の困難な安全性の高い暗号処理が実現される。
【０１３４】
（処理例ａ２）
差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例の第２の例について説明する。図１２のフローチャートを参照して正方ＭＤＳ行列の生成処理について説明する。
【０１３５】
［ステップＳ２０１］
入力：必要なＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、
ＧＦ（2^n）上で、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑをランダムに生成する。なお、図１２に示すフローでは、ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。
【０１３６】
［ステップＳ２０２］
ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑに含まれるｑｍ個の列の任意のｍ個を取り出したときに、正方ＭＤＳ行列になっているかどうかをチェックする。チェックに通過したらステップＳ２０３に進む、そうでない場合はステップＳ２０１にもどる。
なお、正方ＭＤＳ行列とは、前述したように以下の性質を満たす行列をいう。
（ａ）正方行列である

（ｂ）行列に含まれるすべての部分行列（ｓｕｂｍａｔｒｉｘ）の行列式（ｄｅｔｅｒｍｉｎａｎｔ）が０でない、すなわち、ｄｅｔ（ｓｕｂｍａｔｒｉｘ）≠０

［ステップＳ２０３］

ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。

【０１３７】

以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。

【０１３８】

前述の処理例ａ１における正方ＭＤＳ行列生成処理においては、図１１の処理シーケンスにおいて説明したように、ステップＳ１０２において、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑに含まれるｑｍ個の列の任意のｍ個を取り出したときの線形独立性を判定したが、この処理例ａ２における正方ＭＤＳ行列生成処理においては、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑに含まれるｑｍ個の列の任意のｍ個を取り出したとき正方ＭＤＳ行列になっているかどうかをチェックする。すなわち、より厳しいチェックが実行されることになる。

【０１３９】

この図１２に示す処理シーケンスに従った正方ＭＤＳ行列生成処理によって生成されたｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが、先に説明した処理例ａ１と同様、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定される。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。

【０１４０】

このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。

【０１４１】

本構成により、

（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、

（ｂ）暗号化関数内の奇数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列であること、

（ｃ）偶数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列であること、

これら（ａ）～（ｃ）が保証されるため、複数段のラウンド数を持つＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成において、連続する２ｑ－１ラウンドにおいて、ｍ個以下のアクティブＳボックスの寄与による同時差分キャンセルは発生しないことが保証される。

さらに、

（ｄ）正方ＭＤＳの性質から、ａ個（ａ≦ｍ）のアクティブＳボックスの寄与によって得られる差分値における非ゼロの要素数はｍ＋１－ａ個以上になることが保証される。よって暗号化関数全体のアクティブＳボックス数の最小値が増大する。

【０１４２】

このように、本処理例によって、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつである暗号化関数全体でのアクティブＳボックスの最少数を大きくすることが可能となり、結果として、差分解析（差分攻撃）を行なった場合のアクティブＳボックスの数が増大し、解析の困難性が高まることになる。従って鍵の解析の困難な安全性の高い暗号処理が実現される。

【０１４３】
（処理例ａ３）
差分攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例の第３の例について説明する。図１３のフローチャートを参照して正方ＭＤＳ行列の生成処理について説明する。
【０１４４】
［ステップＳ３０１］
入力：必要なＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、
ＧＦ（２^{n}）上で、１個のｑｍ次正方ＭＤＳ行列Ｍを生成する。なお、図１３に示すフローでは、ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。
【０１４５】
［ステップＳ３０２］
１個のｑｍ次正方ＭＤＳ行列ＭからｍＸの行を任意に選択抽出し、ｍ行，ｑｍ列の行列Ｍ'を構成する。
［ステップＳ３０３］
ｍ行，ｑｍ列の行列Ｍ'に含まれるｑｍ本の列ベクトルを重複することなくｍ本の列ベクトルからなるｑ個のグループに任意に分割し、それぞれのグループに含まれる列ベクトルからｍ次の正方行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。
【０１４６】
以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。
【０１４７】
処理例ａ３における正方ＭＤＳ行列生成手法３について、図１４を参照して、より具体的に説明する。
［ステップＳ３０１］
ＧＦ（２^{n}）上で、１個のｑｍ次正方ＭＤＳ行列Ｍを生成する。図１４に示すように、ｑｍ×ｑｍの正方ＭＤＳ行列Ｍを生成する。なお、このステップＳ３０１において生成する行列Ｍの次数はｑｍ次より大きいものでもよい。
［ステップＳ３０２］
図１４に示すように、ｑｍ次正方ＭＤＳ行列ＭからｍＸの行を任意に選択抽出し、ｍ行，ｑｍ列の行列Ｍ'を構成する。なお、図に示す例では、連続するｍ本の行を選択抽出した例として示してあるが、ｍ次正方ＭＤＳ行列Ｍを構成する任意の離間した行をｍ本選択抽出して、ｍ行，ｑｍ列の行列Ｍ'を構成してもよい。
［ステップＳ３０３］
ｍ行，ｑｍ列の行列Ｍ'に含まれるｑｍ本の列ベクトルを重複することなくｍ本の列ベクトルからなるｘ個のグループに任意に分割し、それぞれのグループに含まれる列ベクトルからｍ次の正方行列Ｌ１，Ｌ２，...，Ｌｘを生成する。
【０１４８】
図１３、図１４を参照して説明した処理シーケンスに従った正方ＭＤＳ行列生成処理によって生成されたｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが、先に説明した処理例ａ１、ａ２と同様、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定される。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。
【０１４９】

このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。

【０１５０】

本構成により、

（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、

（ｂ）暗号化関数内の奇数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが独立であること、

（ｃ）偶数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが独立であること、

これら（ａ）～（ｃ）が保証されるため、複数段のラウンド数を持つＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成において、連続する２ｑ－１ラウンドにおいて、ｍ個以下のアクティブＳボックスの寄与による同時差分キャンセルは発生しないことが保証される。

さらに、

（ｄ）正方ＭＤＳの性質から、ａ個（ａ≦ｍ）のアクティブＳボックスの寄与によって得られる差分値における非ゼロの要素数はｍ＋１－ａ個以上になることが保証される。よって暗号化関数全体のアクティブＳボックス数の最小値が増大する。

【０１５１】

なお、処理例ａ３が特に効果を発揮するのは、ｍ，ｒ，が大きくなり、前述した処理例ａ１，ａ２の行列決定処理方式にかかる時間的コストが莫大となり、現実的な時間内に行列を決定することが困難である場合である。そのような場合でも本処理例ａ３の正方ＭＤＳ行列生成手法ならば比較的短時間での行列生成処理が可能となる。

【０１５２】

これは、処理例ａ３においては、大きなｍ，ｒに対しても現実的な時間で十分に処理可能な方式、例えばリードソロモン（Ｒｅｅｄ－Ｓｏｌｏｍｏｎ）符号の生成行列の生成法を適用することが可能となるからである。

【０１５３】

この処理例ａ３においても、上述したように、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつである暗号化関数全体でのアクティブＳボックスの最少数を大きくすることが可能となり、結果として、差分解析（差分攻撃）を行なった場合のアクティブＳボックスの数が増大し、解析の困難性が高まることになる。従って鍵の解析の困難な安全性の高い暗号処理が実現される。

【０１５４】

［（３－ｂ）線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例］

次に、線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例として、２つの処理例ｂ１，ｂ２について説明する。

【０１５５】

（処理例ｂ１）

線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例の第１の例について、説明する。図１５に示すフローチャートを参照して正方ＭＤＳ行列の生成処理について説明する。

【０１５６】

［ステップＳ４０１］

入力：必要な正方ＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、

ＧＦ（２ⁿ）上で、ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑをランダムに生成する。なお、図１４に示すフローでは、正方ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。

【０１５７】

［ステップＳ４０２］

ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑの逆行列Ｍ１$^{-1}$，Ｍ２$^{-1}$，...，Ｍｑ$^{-1}$を算出し、隣り合う２つの逆行列に含まれる２ｍの行ベクトルから任意のｍ本の行ベクトルを取り出したときに、線形独立になっているかどうかをチェックする。図１５中、^{t}Ｒは、行ベクトルの転置ベクトルを示すものである。チェックに通過したらステップＳ４０３に進む、そうでない場合はステップＳ４０１にもどる。ただし、Ｍ１$^{-1}$とＭｑ$^{-1}$は、隣り合う行列とする。

［ステップＳ４０３］

ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。

【０１５８】

以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。

【０１５９】

このようにして生成したｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定する。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。

【０１６０】

このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。

【０１６１】

本構成により、

（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、

（ｂ）暗号化関数内の奇数ラウンド内に連続して含まれる線形変換行列、および偶数ラウンド内に連続して含まれる線形変換行列の逆行列の任意のｍ個の列ベクトルは独立であること、

が保証される。このことにより、線形攻撃における線形近似による解析困難性を高めることが可能となり、解析の困難性、すなわち鍵の解析の困難な安全性の高い暗号処理が実現される。

【０１６２】

（処理例ｂ２）

線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例の第２の例について、説明する。図１６に示すフローチャートを参照して正方ＭＤＳ行列の生成処理について説明する。

【０１６３】

［ステップＳ５０１］

入力：必要な正方ＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、

ＧＦ（２^{n}）上で、ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑをランダムに生成する。なお、図１６に示すフローでは、正方ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。

【０１６４】

［ステップＳ５０２］

ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑの逆行列Ｍ１$^{-1}$，Ｍ２$^{-1}$，...，Ｍｑ$^{-1}$を算出し、隣り合う２つの逆行列に含まれる２ｍの行ベクトルから任意のｍ本の行ベクトルを取り出したときに、正方ＭＤＳ行列になっているかどうかをチェックする。図１６中、^{t}Ｒは、行ベクトルの転置ベクトルを示すものである。チェックに通過したら

ステップＳ５０３に進む、そうでない場合はステップＳ４０１にもどる。ただし、Ｍｌ－１とＭｑ－１は、隣り合う行列とする。

なお、正方ＭＤＳ行列とは、前述したように以下の性質を満たす行列をいう。

（ａ）正方行列である

（ｂ）行列に含まれるすべての部分行列（ｓｕｂｍａｔｒｉｘ）の行列式（ｄｅｔｅｒｍｉｎａｎｔ）が０でない、すなわち、ｄｅｔ（ｓｕｂｍａｔｒｉｘ）≠０

【０１６５】

［ステップＳ５０３］

ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。

【０１６６】

以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。

【０１６７】

前述の処理例ｂ１における正方ＭＤＳ行列生成処理においては、図１５の処理シーケンスにおいて説明したように、ステップＳ４０２において、ｑ個のｍ次正方ＭＤＳ行列のＭ１，Ｍ２，...，Ｍｑの逆行列Ｍ１－１，Ｍ２－１，...，Ｍｑ－１に含まれるｑｍ個の列の任意のｍ個を取り出したときの線形独立性を判定したが、この処理例ｂ２における正方ＭＤＳ行列生成処理においては、ｑ個のｍ次正方ＭＤＳ行列のＭ１，Ｍ２，...，Ｍｑの逆行列Ｍ１－１，Ｍ２－１，...，Ｍｑ－１に含まれるｑｍ個の列の任意のｍ個を取り出したとき正方ＭＤＳ行列になっているかどうかをチェックする。すなわち、より厳しいチェックが実行されることになる。

【０１６８】

この図１６に示す処理シーケンスに従った正方ＭＤＳ行列生成処理によって生成されたｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが、先に説明した処理例ｂ１と同様、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定される。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。

【０１６９】

このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。

【０１７０】

本構成により、

（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、

（ｂ）暗号化関数内の奇数ラウンド内に連続して含まれる線形変換行列、および偶数ラウンド内に連続して含まれる線形変換行列の逆行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列となること、

が保証される。このことにより、線形攻撃における線形近似による解析困難性を高めることが可能となり、解析の困難性、すなわち鍵の解析の困難な安全性の高い暗号処理が実現される。

【０１７１】

［（３－ｃ）差分攻撃および線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例］

次に、差分攻撃および線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列の生成およびＦ関数への設定例について説明する。

【０１７２】
差分攻撃に対する耐性を持つ暗号処理アルゴリズムは、先に、図１０～図１３を参照して説明した処理、すなわち、Ｆ関数の線形処理部における線形変換に適用する正方ＭＤＳ行列を前述の処理例ａ１（図１１）～ａ３（図１３）のいずれかの処理を適用して設定することで実現される。また、線形攻撃に対する耐性を持つ暗号処理アルゴリズムは、先に、図１０、および図１４、図１５を参照して説明した処理、すなわち、Ｆ関数の線形処理部における線形変換に適用する正方ＭＤＳ行列を前述の処理例ｂ１（図１４）、ｂ２（図１５）のいずれかの処理を適用して設定することで実現される。
【０１７３】
差分攻撃および線形攻撃に対する耐性向上を実現した正方ＭＤＳ行列は、
処理例ａ１（図１１）～ａ３（図１３）のいずれかの処理と、
処理例ｂ１（図１４）、ｂ２（図１５）のいずれかの処理とを、
併せて実行して生成した正方ＭＤＳ行列を、図１０において説明した［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定することで実現される。
【０１７４】
すなわち、
処理例ａ１と処理例ｂ１、
処理例ａ１と処理例ｂ２、
処理例ａ２と処理例ｂ１、
処理例ａ２と処理例ｂ２、
処理例ａ３と処理例ｂ１、
処理例ａ３と処理例ｂ２、
のいずれかの組み合わせによって、ｑ個の正方ＭＤＳ行列を生成し、２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定する。奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。この設定により、差分攻撃および線形攻撃に対する耐性向上を実現した暗号処理が可能となる。
【０１７５】
図１７を参照して、差分攻撃および線形攻撃に対する耐性向上を実現した暗号処理を実現するための正方ＭＤＳ行列の生成処理の一例について説明する。この処理は、前述した処理例ａ２と、処理例ｂ２との組み合わせである。
【０１７６】
［ステップＳ６０１］
入力：必要な正方ＭＤＳの個数ｑ，拡大次数：ｎ，行列のサイズ：ｍとして、
ＧＦ（2^n）上で、ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑをランダムに生成する。なお、図１７に示すフローでは、正方ＭＤＳの個数ｑ＝６，拡大次数：ｎ＝８，行列のサイズ：ｍ＝８の場合の処理例として示してある。
【０１７７】
［ステップＳ６０２］
ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑに含まれるｑｍ個の列の任意のｍ個を取り出したときに、正方ＭＤＳ行列になっているかどうかをチェックする。チェックに通過したらステップＳ６０３に進む、そうでない場合はステップＳ６０１にもどる。
なお、正方ＭＤＳ行列とは、前述したように以下の性質を満たす行列をいう。
（ａ）正方行列である
（ｂ）行列に含まれるすべての部分行列（ｓｕｂｍａｔｒｉｘ）の行列式（ｄｅｔｅｒｍｉｎａｎｔ）が０でない、すなわち、ｄｅｔ（ｓｕｂｍａｔｒｉｘ）≠０

【０１７８】
［ステップＳ６０３］
ｑ個のｍ次正方ＭＤＳ行列Ｍ１，Ｍ２，...，Ｍｑの逆行列Ｍ１－１，Ｍ２－１，...，Ｍｑ－１を算出し、隣り合う２つの逆行列に含まれる２ｍの行ベクトルから任意のｍ本の行ベクトルを取り出したときに、正方ＭＤＳ行列になっているかどうかをチェックする。図１７中、ｔＲは、行ベクトルの転置ベクトルを示すものである。チェックに通過したらステップＳ６０４に進む、そうでない場合はステップＳ６０１にもどる。ただし、Ｍ１－１とＭｑ－１は、隣り合う行列とする。
【０１７９】
［ステップＳ６０４］
ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑを、ラウンド数２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号に適用する正方ＭＤＳ行列として出力する。
【０１８０】
以上のプロセスによって、ｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが生成される。なお、ｑ≦ｒである。
【０１８１】
この図１７に示す処理シーケンスに従った正方ＭＤＳ行列生成処理によって生成されたｑ個のｍ次正方ＭＤＳ行列Ｌ１，Ｌ２，...，Ｌｑが、先に、図１０を参照して説明した、［ステップＳ２３］、［ステップＳ２４］の処理に従って、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として設定される。すなわち、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定する。
【０１８２】
このように、偶数ラウンドの正方ＭＤＳ行列と奇数ラウンドの正方ＭＤＳ行列を互いに逆順に配置することによって、暗号化処理と復号処理は鍵の順序を入れ替える処理を除き同一であることが保証される。
【０１８３】
本構成により、
（ａ）各Ｆ関数の線形変換行列は正方ＭＤＳであること、
（ｂ）暗号化関数内の奇数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列であること、
（ｃ）偶数ラウンド内の少なくとも連続するｑ個のＦ関数に含まれる線形変換行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列であること、
これら（ａ）～（ｃ）が保証されるため、複数段のラウンド数を持つＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成において、連続する２ｑ－１ラウンドにおいて、ｍ個以下のアクティブＳボックスの寄与による同時差分キャンセルは発生しないことが保証される。
さらに、
（ｄ）正方ＭＤＳの性質から、ａ個（ａ≦ｍ）のアクティブＳボックスの寄与によって得られる差分値における非ゼロの要素数はｍ＋１－ａ個以上になることが保証される。よって暗号化関数全体のアクティブＳボックス数の最小値が増大する。
さらに、
（ｅ）暗号化関数内の奇数ラウンド内に連続して含まれる線形変換行列、および偶数ラウンド内に連続して含まれる線形変換行列の逆行列の任意のｍ個の列ベクトルが正方ＭＤＳ行列となることが保証される。このことにより、線形攻撃における線形近似による解析困難性を高めることが可能となり、解析の困難性、すなわち鍵の解析の困難な安全性の高い暗号処理が実現される。
【０１８４】
このように、本処理例によって、差分攻撃および線形攻撃の双方の解析の困難性が向上

し、鍵の解析の困難な安全性の高い暗号処理が実現される。なお、図１７に示した例は、前述したように、先に説明した処理例ａ２と処理例ｂ２の組み合わせによる正方ＭＤＳ行列の生成例であるが、その他、処理例ａ１と処理例ｂ１、処理例ａ１と処理例ｂ２、処理例ａ２と処理例ｂ１、処理例ａ３と処理例ｂ１、処理例ａ３と処理例ｂ２とを組み合わせてｑ個の正方ＭＤＳ行列の生成を行い、段数（ラウンド数）が２ｒのＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理構成の各段のＦ関数部の線形変換部の線形変換処理に適用する行列として、奇数段目については上位段から順にＬ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・としてｑ個の正方ＭＤＳ行列を繰り返し設定し、偶数段のＦ関数については、下位段のＦ関数から順に、Ｌ１，Ｌ２，・・，Ｌｑ，Ｌ１，Ｌ２・・として、ｑ個の正方ＭＤＳ行列を繰り返し設定することによっても、差分攻撃および線形攻撃の双方の解析の困難性が高く、鍵の解析の困難な安全性の高い暗号処理が実現可能である。

【０１８５】

これまでの説明では、分かりやすさを優先して線形変換行列をＧＦ（２^{n}）上で定義されるｍ×ｍの行列として、ｍｎビットからｍｎビットへのデータ変換演算としてきたが、差分解析および線形解析に対する同様な効果がＧＦ（２）上で定義されるｍｎ×ｍｎの行列を用いた場合でも有効である。実際、任意のＧＦ（２^{n}）上の行列は同じ変換を表すＧＦ（２）上の行列に一対一で対応させることができる。よってＧＦ（２）上の行列はより一般的な表現を表しているといえる。ＧＦ（２）上では行と列の本数がｍｎ本ずつと、ＧＦ（２^{n}）の場合と比べてｎ倍となる。このためＧＦ（２^{n}）上の行列の１行目は、ＧＦ（２）上の行列の１からｎ行目に対応し、1列目は１からｎ列目に対応している。つまりｉ行目は（ｉ－１）＋１行目から（ｉ－１）＋ｎ行目に対応し、ｉ列目は（ｉ－１）＋１列目から（ｉ－１）＋ｎ列目に対応している。よって、ＧＦ（２^{n}）上の行や列を取り出してくる操作には、ＧＦ（２）上で定義される行列を用いる場合は、対応するｎ行分もしくはｎ列分を取り出すという操作を対応させればよい。ＧＦ（２^{n}）上のｍ本の行や列を取り出す操作は、ＧＦ（２）上ではｎ本の行や列をｍ回取り出すことになり、結果としてｍｎ×ｍｎの行列を得ることができる。以上の対応付けにより、ＧＦ（２）上で定義される行列に容易に拡張することができる。

【０１８６】

最後に、暗号処理を実行する暗号処理装置としてのＩＣモジュール６００の構成例を図１８に示す。上述の処理は、例えばＰＣ、ＩＣカード、リーダライタ、その他、様々な情報処理装置において実行可能であり、図１８に示すＩＣモジュール６００は、これら様々な機器に構成することが可能である。

【０１８７】

図１８に示すＣＰＵ(Central processing Unit)６０１は、暗号処理の開始や、終了、データの送受信の制御、各構成部間のデータ転送制御、その他の各種プログラムを実行するプロセッサである。メモリ６０２は、ＣＰＵ６０１が実行するプログラム、あるいは演算パラメータとしての固定データを格納するＲＯＭ（Read-Only-Memory）、ＣＰＵ６０１の処理において実行されるプログラム、およびプログラム処理において適宜変化するパラメータの格納エリア、ワーク領域として使用されるＲＡＭ（Random Access Memory）等からなる。また、メモリ６０２は暗号処理に必要な鍵データ等の格納領域として使用可能である。データ等の格納領域は、耐タンパ構造を持つメモリとして構成されることが好ましい。

【０１８８】

暗号処理部６０３は、例えば上述したＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理アルゴリズムに従った暗号処理、復号処理等を実行する。なお、ここでは、暗号処理手段を個別モジュールとした例を示したが、このような独立した暗号処理モジュールを設けず、例えば暗号処理プログラムをＲＯＭに格納し、ＣＰＵ６０１がＲＯＭ格納プログラムを読み出して実行するように構成してもよい。

【０１８９】

乱数発生器６０４は、暗号処理に必要となる鍵の生成などにおいて必要となる乱数の発

生処理を実行する。

【０１９０】

送受信部６０５は、外部とのデータ通信を実行するデータ通信処理部であり、例えばリーダライタ等、ＩＣモジュールとのデータ通信を実行し、ＩＣモジュール内で生成した暗号文の出力、あるいは外部のリーダライタ等の機器からのデータ入力などを実行する。

【０１９１】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、特許請求の範囲の欄を参酌すべきである。

【０１９２】

なお、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。

【０１９３】

例えば、プログラムは記録媒体としてのハードディスクやＲＯＭ（Read Only Memory）に予め記録しておくことができる。あるいは、プログラムはフレキシブルディスク、ＣＤ－ＲＯＭ(Compact Disc Read Only Memory)，ＭＯ(Magneto optical)ディスク，ＤＶＤ(Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体は、いわゆるパッケージソフトウエアとして提供することができる。

【０１９４】

なお、プログラムは、上述したようなリムーバブル記録媒体からコンピュータにインストールする他、ダウンロードサイトから、コンピュータに無線転送したり、ＬＡＮ(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

【０１９５】

なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【産業上の利用可能性】

【０１９６】

上述したように、本発明の構成によれば、非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理において、複数ラウンド各々に対応するＦ関数の線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行するとともに、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列：Ｌａ，Ｌｂを適用し、かつ該正方ＭＤＳ行列の逆行列：Ｌａ$^{-1}$，Ｌｂ$^{-1}$を構成する列ベクトルから任意に選択したｍ個の列ベクトルによって構成する行列が線形独立であるか、あるいは正方ＭＤＳ行列を構成する性質とした正方ＭＤＳ行列による線形変換処理を実行する構成としたので、共通鍵ブロック暗号における線形攻撃に対する耐性が向上し、暗号鍵等の解析困難性が高まることとなり、安全性の高い暗号処理が実現される。従って、鍵解析困難性を高め、安全性の要求される暗号処理実行装置において適用可能である。

【０１９７】
さらに、本発明の構成によれば、非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理において、複数ラウンド各々に対応するＦ関数の線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ Ｄｉｓｔａｎｃｅ Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理として実行するとともに、少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において異なる正方ＭＤＳ行列を適用する構成とし、これらの正方ＭＤＳ行列自身が、線形独立性を示すか、または正方ＭＤＳ行列を構成する構成としたので、アクティブＳボックスの寄与による同時差分キャンセルの発生しないことが保証され、共通鍵ブロック暗号における差分攻撃に対する強度指標のひとつである暗号化関数全体でのアクティブＳボックスの最少数を大きくすることが可能となる。本構成により、線形攻撃、差分攻撃の双方に対して耐性が向上し、より安全性の高い暗号処理が実現される。従って、鍵解析困難性を高め、安全性の要求される暗号処理実行装置において適用可能である。

【図面の簡単な説明】

【０１９８】
【図１】Ｆｅｉｓｔｅｌ構造を持つ代表的な共通鍵ブロック暗号の構成を示す図である。
【図２】ラウンド関数部として設定されるＦ関数の構成について説明する図である。
【図３】線形変換部において、線形変換処理に適用する正方行列の例を示す図である。
【図４】ｍ＝８，ｎ＝８の１２８ｂｉｔブロック暗号における３段の同時差分キャンセルの様子を説明する図である。
【図５】Ｆ関数の線形変換部において、正方行列による線形変換が実行されて、Ｆ関数出力差分ΔＹｉを生成する具体例を説明する図である。
【図６】ｍ＝８，ｎ＝８の１２８ｂｉｔブロック暗号における５段の同時差分キャンセルの様子を説明する図である。
【図７】共通鍵ブロック暗号における任意段の同時差分キャンセルの定義を説明する図である。
【図８】正方ＭＤＳ行列の一例を示す図である。
【図９】本発明に係る共通鍵ブロック暗号処理アルゴリズムにおける各ラウンドのＦ関数の線形変換行列としての正方ＭＤＳ行列設定例を説明する図である。
【図１０】本発明に係る共通鍵ブロック暗号処理アルゴリズムにおける各ラウンドのＦ関数の線形変換行列としての正方ＭＤＳ行列設定処理シーケンスを説明するフロー図である。
【図１１】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、差分攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例ａ１を説明するフロー図である。
【図１２】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、差分攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例ａ２を説明するフロー図である。
【図１３】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、差分攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例ａ３を説明するフロー図である。
【図１４】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成処理例ａ３の具体的手法を説明する図である。
【図１５】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、線形攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例ｂ１を説明するフロー図である。
【図１６】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、線形攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例ｂ２を

説明するフロー図である。

【図１７】各ラウンドのＦ関数に設定する線形変換行列である正方ＭＤＳ行列の生成手法として、差分攻撃および線形攻撃に対する耐性向上を実現する正方ＭＤＳ行列生成処理例を説明するフロー図である。

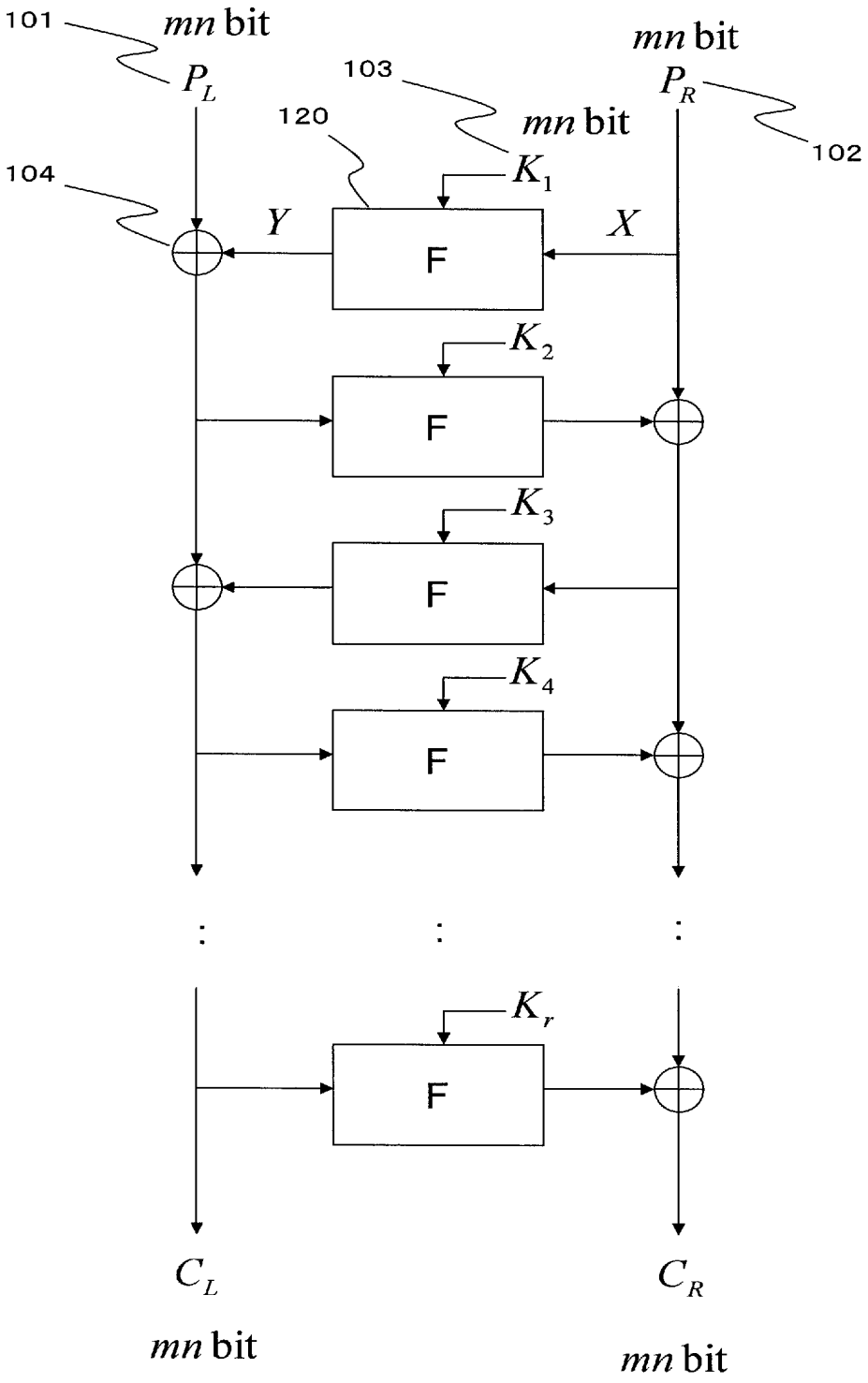
【図１８】本発明にかかる暗号処理を実行する暗号処理装置としてのＩＣモジュールの構成例を示す図である。

【符号の説明】

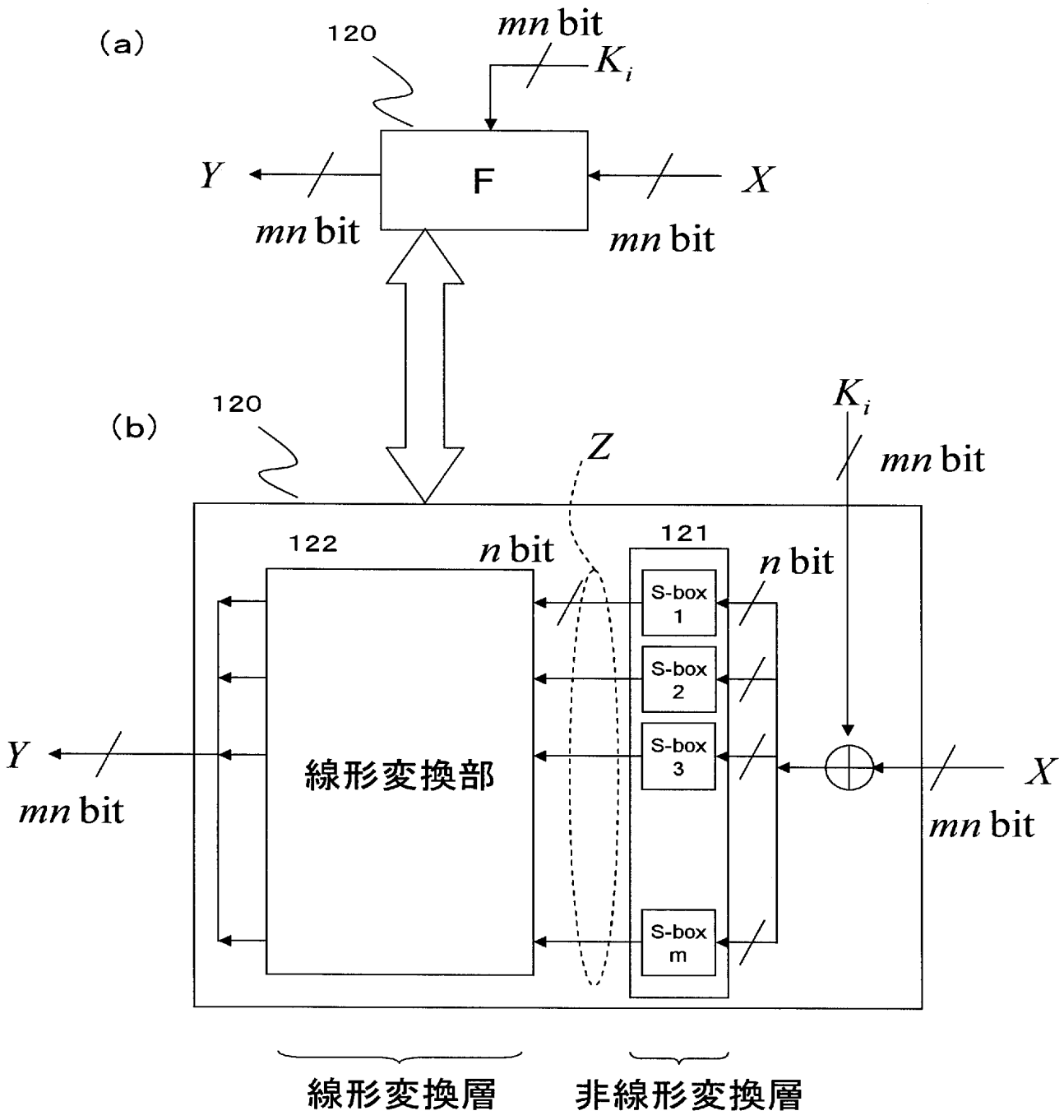
【０１９９】

１０１　入力データP_L（Plain-Left）
１０２　入力データP_R(Plain-Right)
１０３　ラウンド鍵
１０４　排他的論理和部
１２０　Ｆ関数
１２１　Ｓボックス（Ｓ－ｂｏｘ）
１２２　線形変換部
１２５　正方行列
１３１，１３３　排他的論理和部
１４１～１４５　排他的論理和部
４０１　Ｆ関数
４０２　正方ＭＤＳ行列
６００　ＩＣモジュール
６０１　ＣＰＵ(Central processing Unit)
６０２　メモリ
６０３　暗号処理部
６０４　乱数発生器
６０５　送受信部

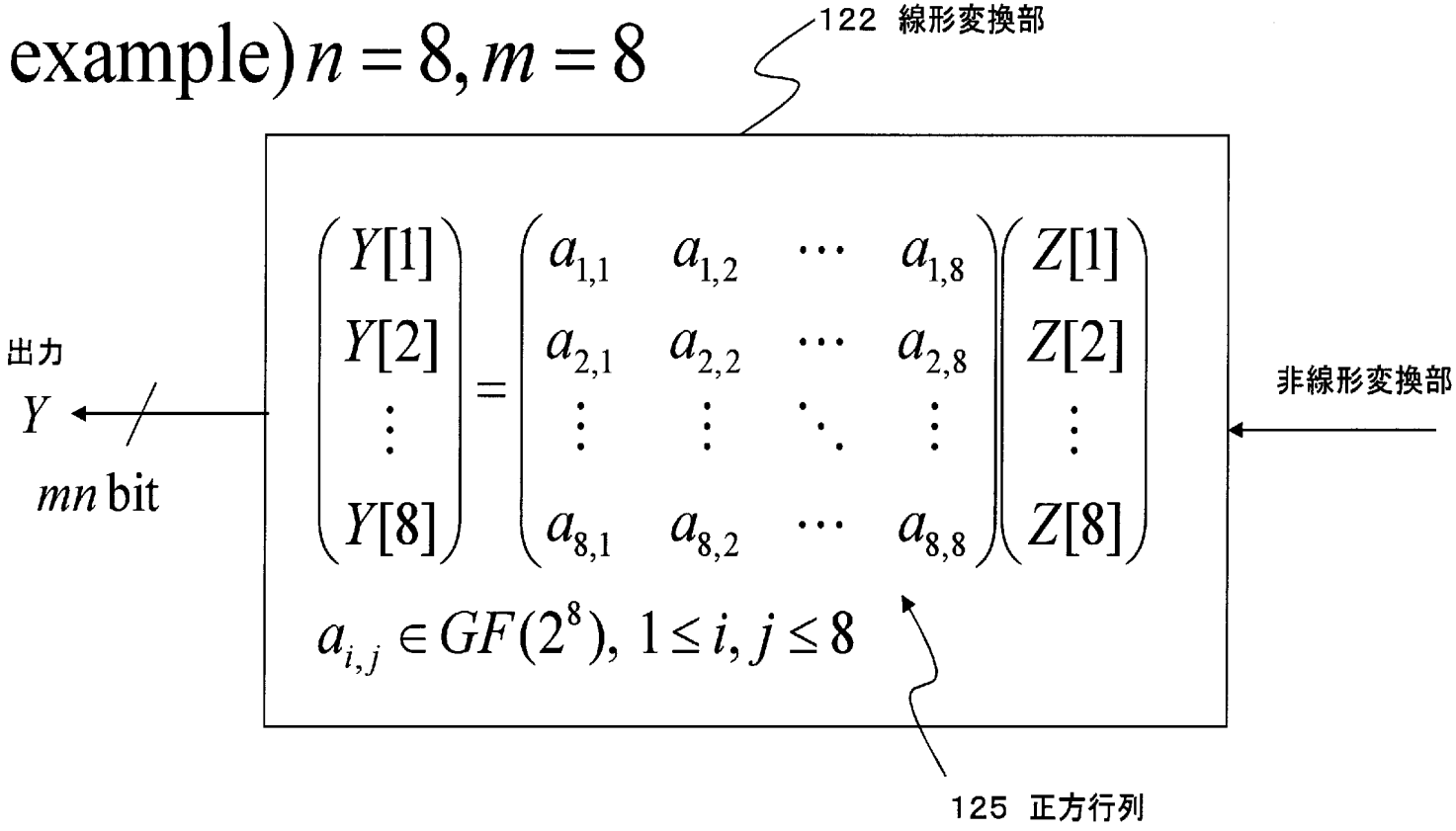
【書類名】図面
【図１】


101
mn bit
P_L
103
mn bit
P_R
102
120
mn bit
K_1
104
Y
X
F
K_2
F
K_3
F
K_4
F
K_r
F
C_L
mn bit
C_R
mn bit

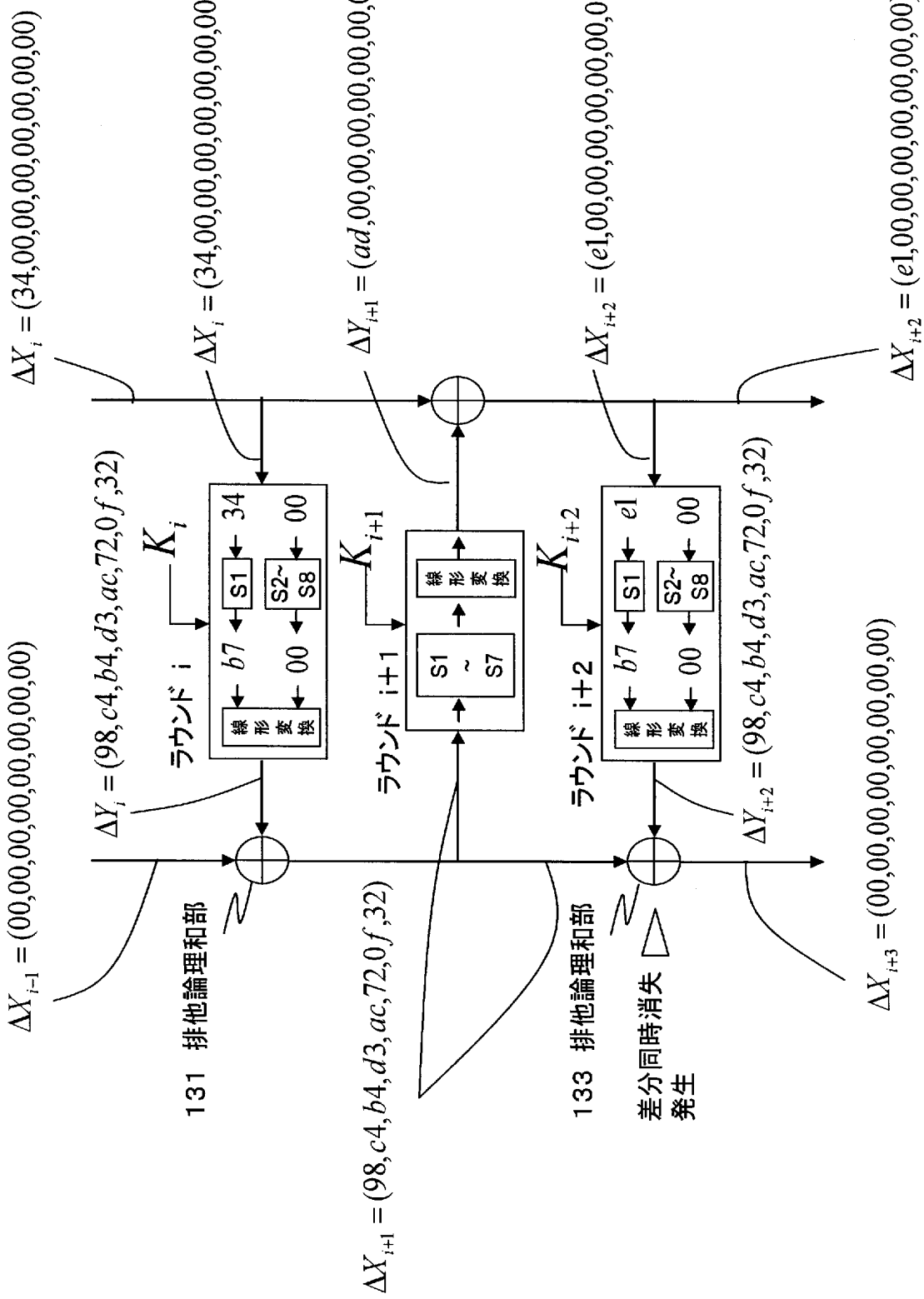
【図２】


(a)
120
mn bit
K_i
Y
F
X
mn bit
mn bit
(b)
120
K_i
Z
mn bit
122
n bit
121
n bit
S-box 1
S-box 2
S-box 3
S-box m
Y
線形変換部
X
mn bit
mn bit
線形変換層
非線形変換層

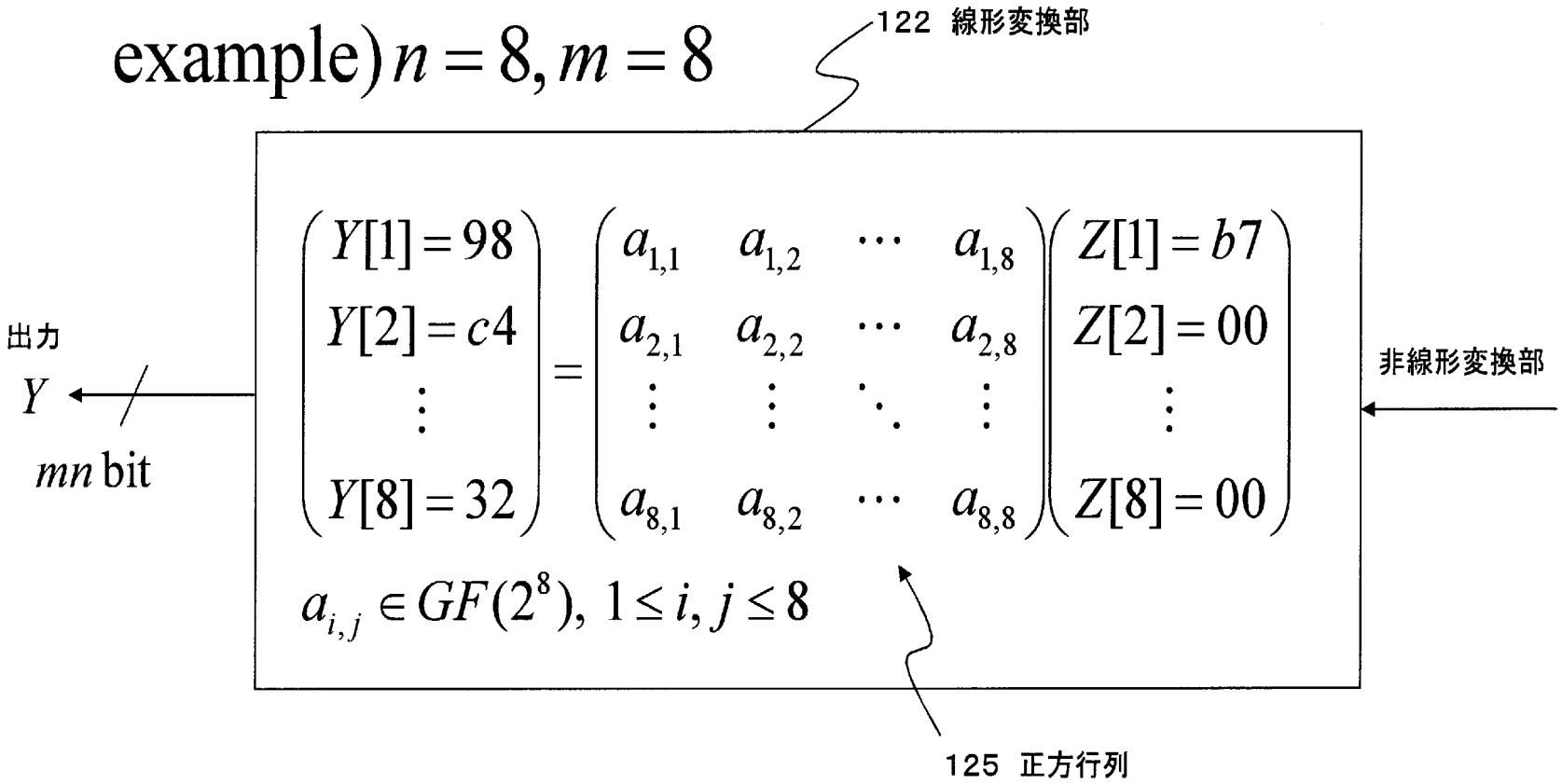
【図３】


example) $n=8, m=8$
122 線形変換部
$\begin{pmatrix} Y[1] \\ Y[2] \\ \vdots \\ Y[8] \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,8} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ a_{8,1} & a_{8,2} & \cdots & a_{8,8} \end{pmatrix} \begin{pmatrix} Z[1] \\ Z[2] \\ \vdots \\ Z[8] \end{pmatrix}$
$a_{i,j} \in GF(2^8),\ 1 \le i, j \le 8$
出力
Y
mn bit
非線形変換部
125 正方行列

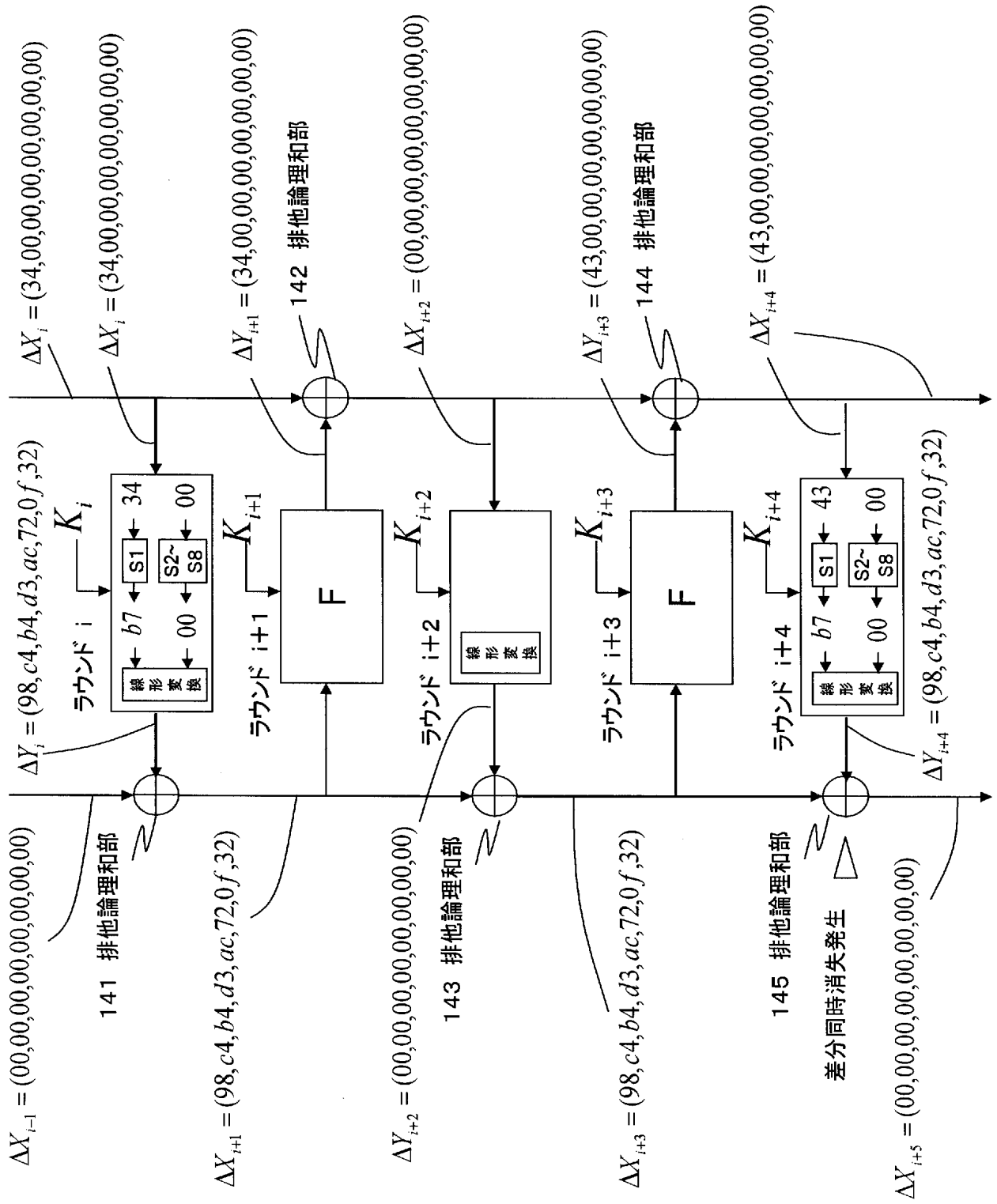
【図４】


$\Delta X_i = (34,00,00,00,00,00,00,00)$
$\Delta X_i = (34,00,00,00,00,00,00,00)$
$\Delta Y_{i+1} = (ad,00,00,00,00,00,00,00)$
$\Delta X_{i+2} = (e1,00,00,00,00,00,00,00)$
$\Delta X_{i+2} = (e1,00,00,00,00,00,00,00)$
$\Delta X_{i-1} = (00,00,00,00,00,00,00,00)$
$\Delta Y_i = (98,c4,b4,d3,ac,72,0f,32)$
K_i
ラウンド i
線形変換
b7
00
S1
34
S2~S8
00
131 排他論理和部
$\Delta X_{i+1} = (98,c4,b4,d3,ac,72,0f,32)$
K_{i+1}
ラウンド i+1
S1~S7
線形変換
K_{i+2}
ラウンド i+2
線形変換
b7
00
S1
e1
S2~S8
00
133 排他論理和部
差分同時消失発生
$\Delta Y_{i+2} = (98,c4,b4,d3,ac,72,0f,32)$
$\Delta X_{i+3} = (00,00,00,00,00,00,00,00)$

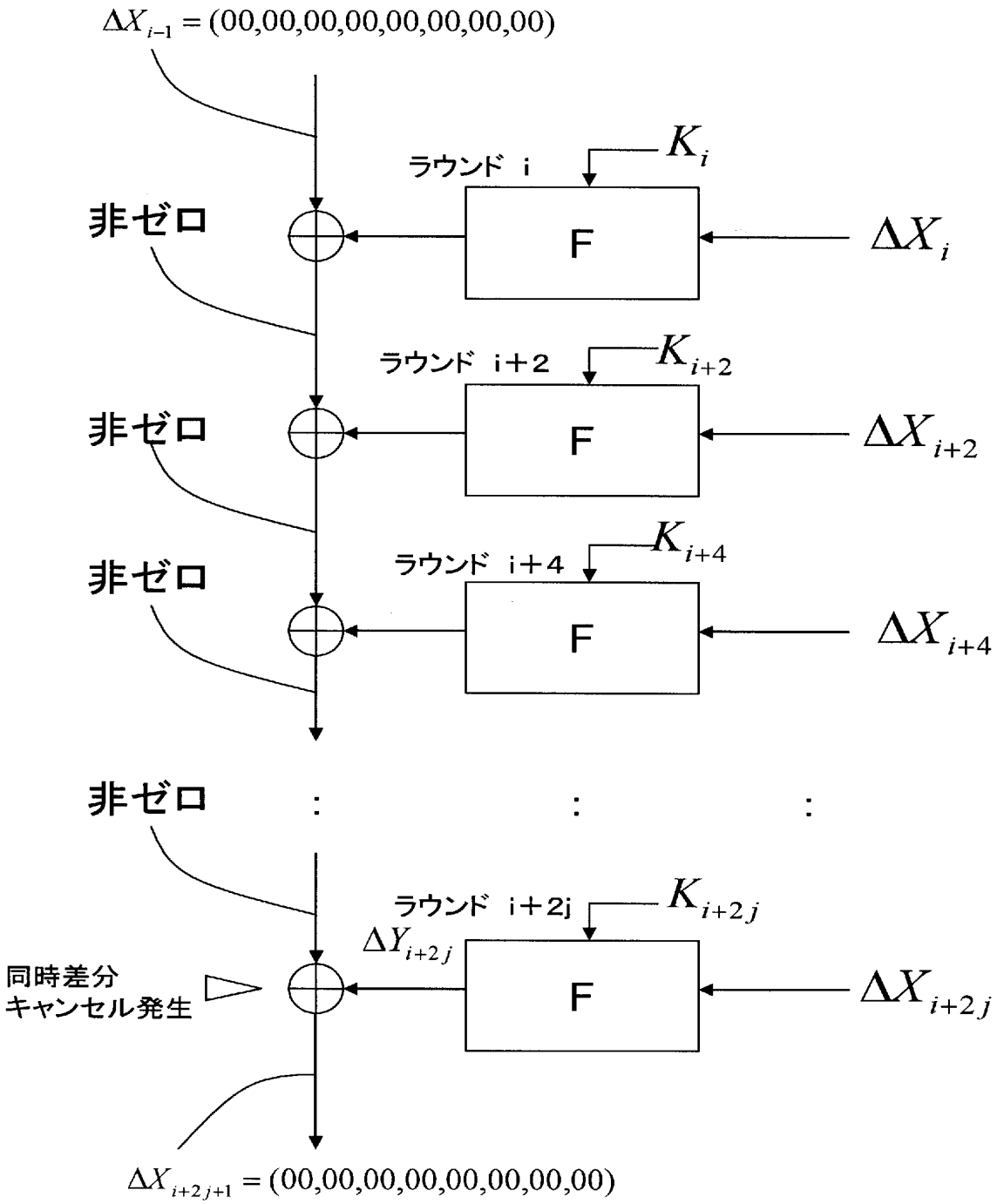
【図５】


example) $n=8, m=8$
122 線形変換部
$\begin{pmatrix} Y[1]=98 \\ Y[2]=c4 \\ \vdots \\ Y[8]=32 \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,8} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,8} \\ \vdots & \vdots & \ddots & \vdots \\ a_{8,1} & a_{8,2} & \cdots & a_{8,8} \end{pmatrix} \begin{pmatrix} Z[1]=b7 \\ Z[2]=00 \\ \vdots \\ Z[8]=00 \end{pmatrix}$
$a_{i,j} \in GF(2^8),\ 1 \le i, j \le 8$
出力
Y
mn bit
非線形変換部
125 正方行列

【図６】


ΔX_{i-1} = (00,00,00,00,00,00,00,00)
ΔY_i = (98,c4,b4,d3,ac,72,0f,32)
ΔX_i = (34,00,00,00,00,00,00,00)
141 排他論理和部
ラウンド i
K_i
線形変換
b7
34
00
S1
S2~S8
ΔX_{i+1} = (98,c4,b4,d3,ac,72,0f,32)
ΔY_{i+1} = (34,00,00,00,00,00,00,00)
142 排他論理和部
ラウンド i+1
K_{i+1}
F
ΔY_{i+2} = (00,00,00,00,00,00,00,00)
143 排他論理和部
ラウンド i+2
K_{i+2}
線形変換
ΔX_{i+2} = (00,00,00,00,00,00,00,00)
ΔX_{i+3} = (98,c4,b4,d3,ac,72,0f,32)
ΔY_{i+3} = (43,00,00,00,00,00,00,00)
144 排他論理和部
ラウンド i+3
K_{i+3}
F
ΔX_{i+4} = (43,00,00,00,00,00,00,00)
145 排他論理和部
差分同時消失発生
ラウンド i+4
K_{i+4}
線形変換
b7
43
00
S1
S2~S8
ΔY_{i+4} = (98,c4,b4,d3,ac,72,0f,32)
ΔX_{i+5} = (00,00,00,00,00,00,00,00)

【図７】

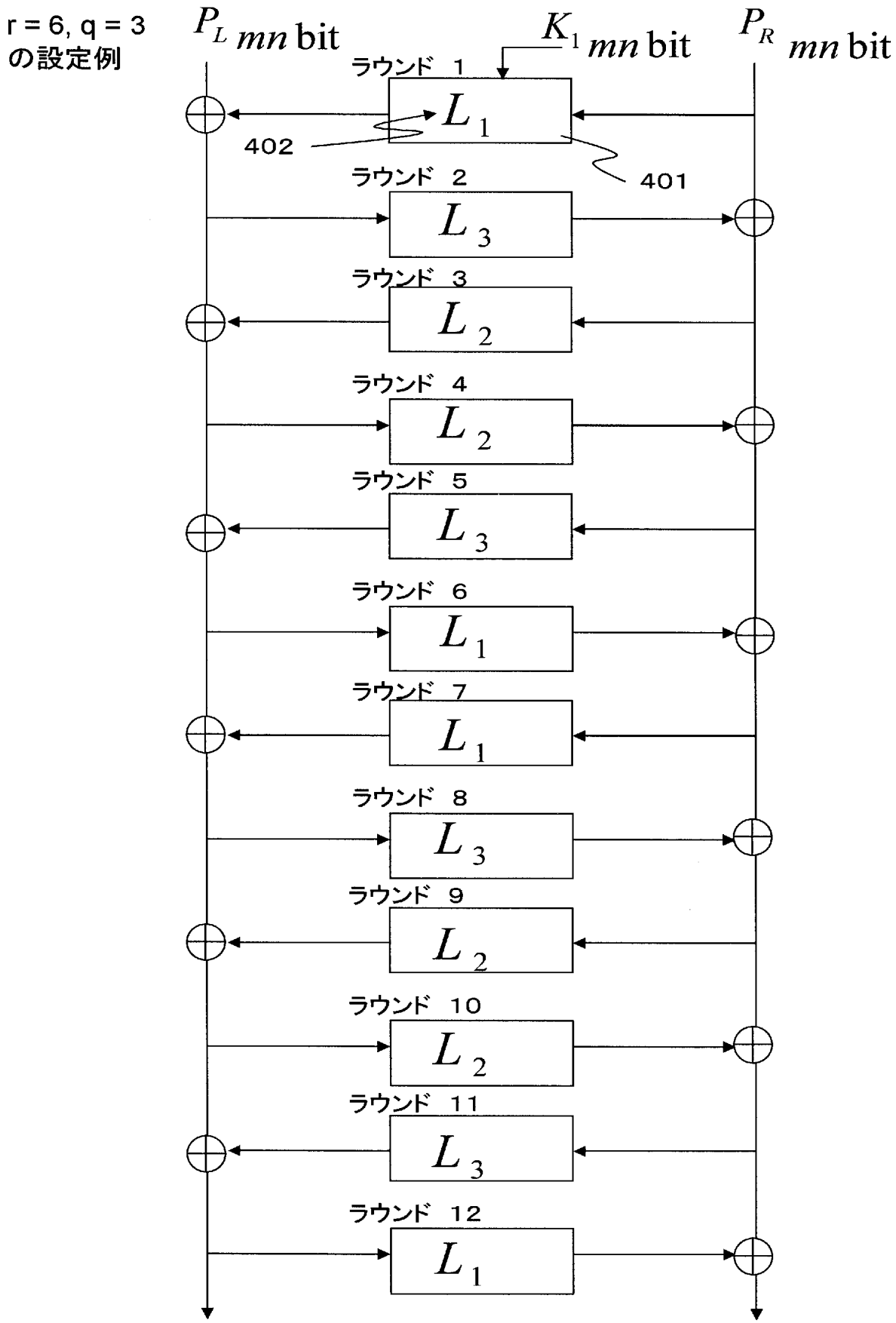

$\Delta X_{i-1} = (00,00,00,00,00,00,00,00)$
ラウンド i
K_i
非ゼロ
F
ΔX_i
ラウンド i＋2
K_{i+2}
非ゼロ
F
ΔX_{i+2}
K_{i+4}
ラウンド i＋4
非ゼロ
F
ΔX_{i+4}
非ゼロ
:
:
:
ラウンド i＋2j
K_{i+2j}
ΔY_{i+2j}
同時差分
キャンセル発生
F
ΔX_{i+2j}
$\Delta X_{i+2j+1} = (00,00,00,00,00,00,00,00)$

【図８】

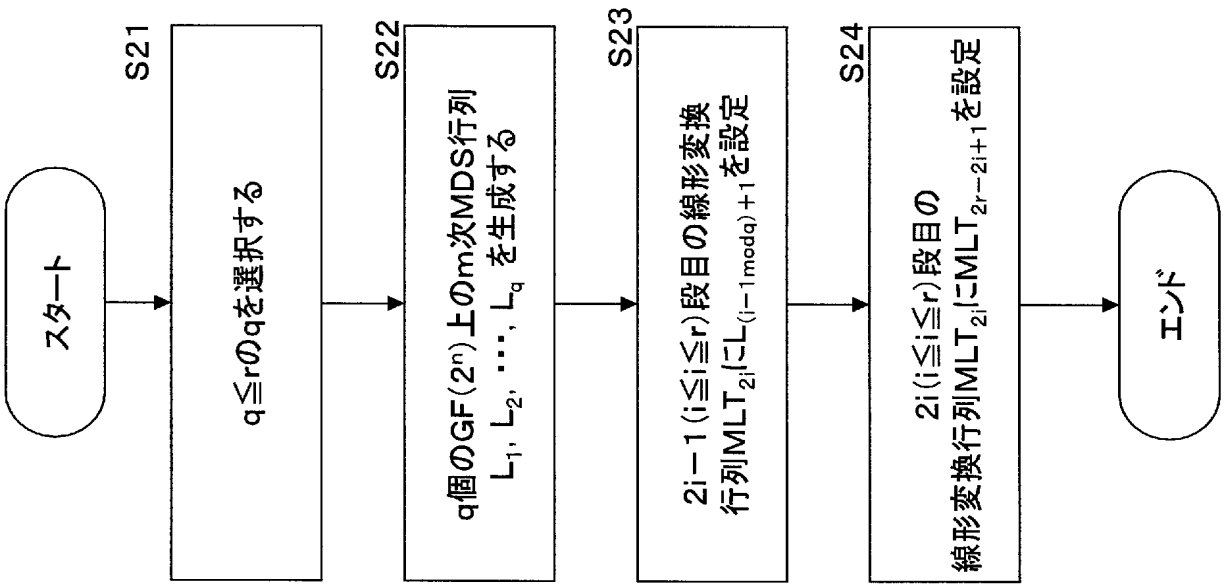
example) $n = 8, m = 8$

$$\begin{pmatrix} 9d & b4 & d3 & 5d & 84 & ae & ec & b9 \\ 29 & 34 & 39 & 60 & 5c & 81 & 25 & 13 \\ 67 & 6a & d2 & e3 & 4b & db & 9d & 4 \\ 8e & d7 & e6 & 1b & 8b & 9e & 3a & 91 \\ d9 & e5 & 4d & dd & c6 & 5 & f0 & ad \\ 2a & f7 & 67 & 72 & b1 & 7 & f2 & 27 \\ 42 & e6 & a0 & 4 & f1 & 4 & 7d & 8c \\ 55 & 63 & fa & 51 & c & d9 & 28 & d6 \end{pmatrix}$$

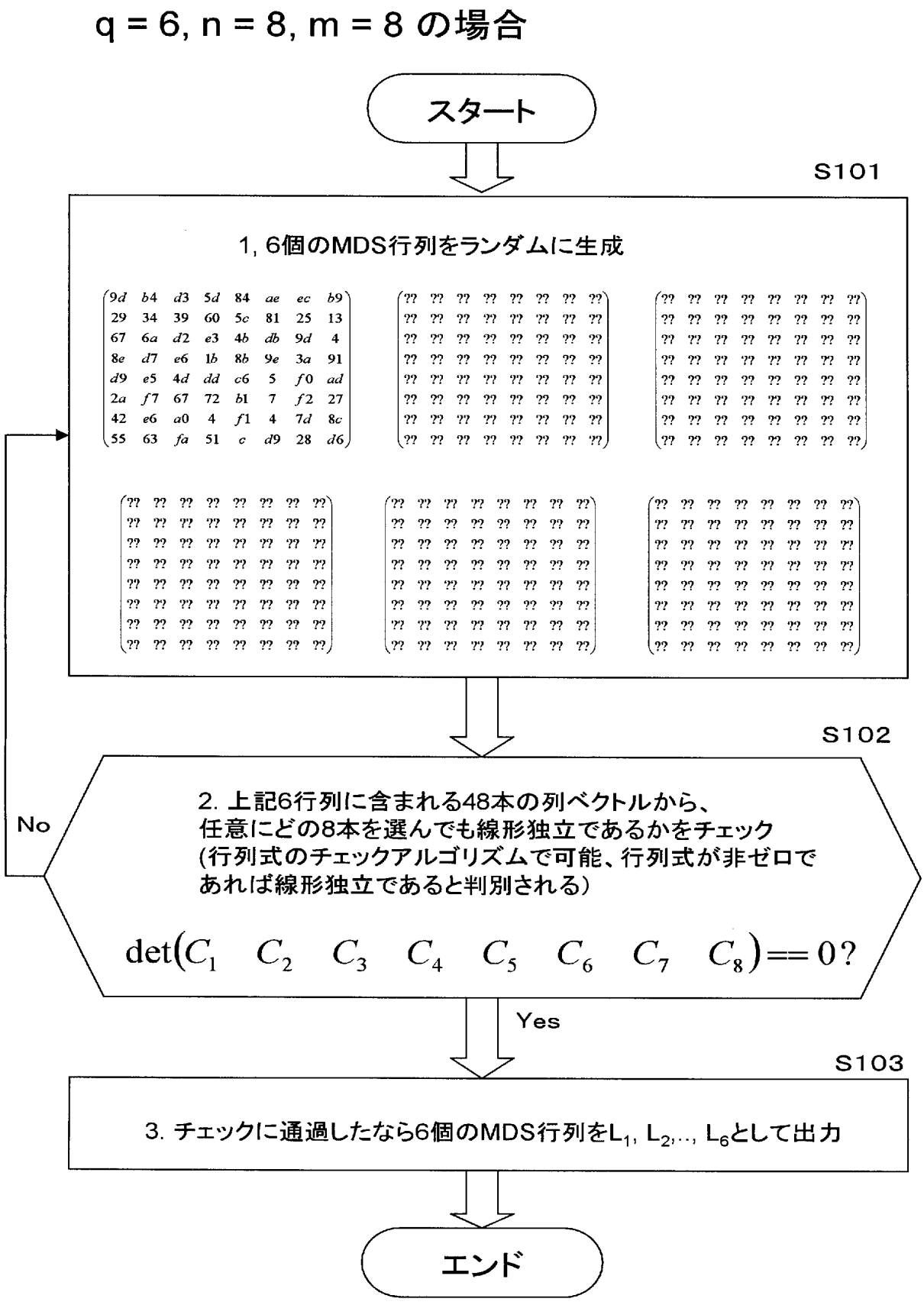
【図９】


r = 6, q = 3
の設定例
P_L mn bit
K_1 mn bit
P_R mn bit
ラウンド 1
L_1
402
401
ラウンド 2
L_3
ラウンド 3
L_2
ラウンド 4
L_2
ラウンド 5
L_3
ラウンド 6
L_1
ラウンド 7
L_1
ラウンド 8
L_3
ラウンド 9
L_2
ラウンド 10
L_2
ラウンド 11
L_3
ラウンド 12
L_1

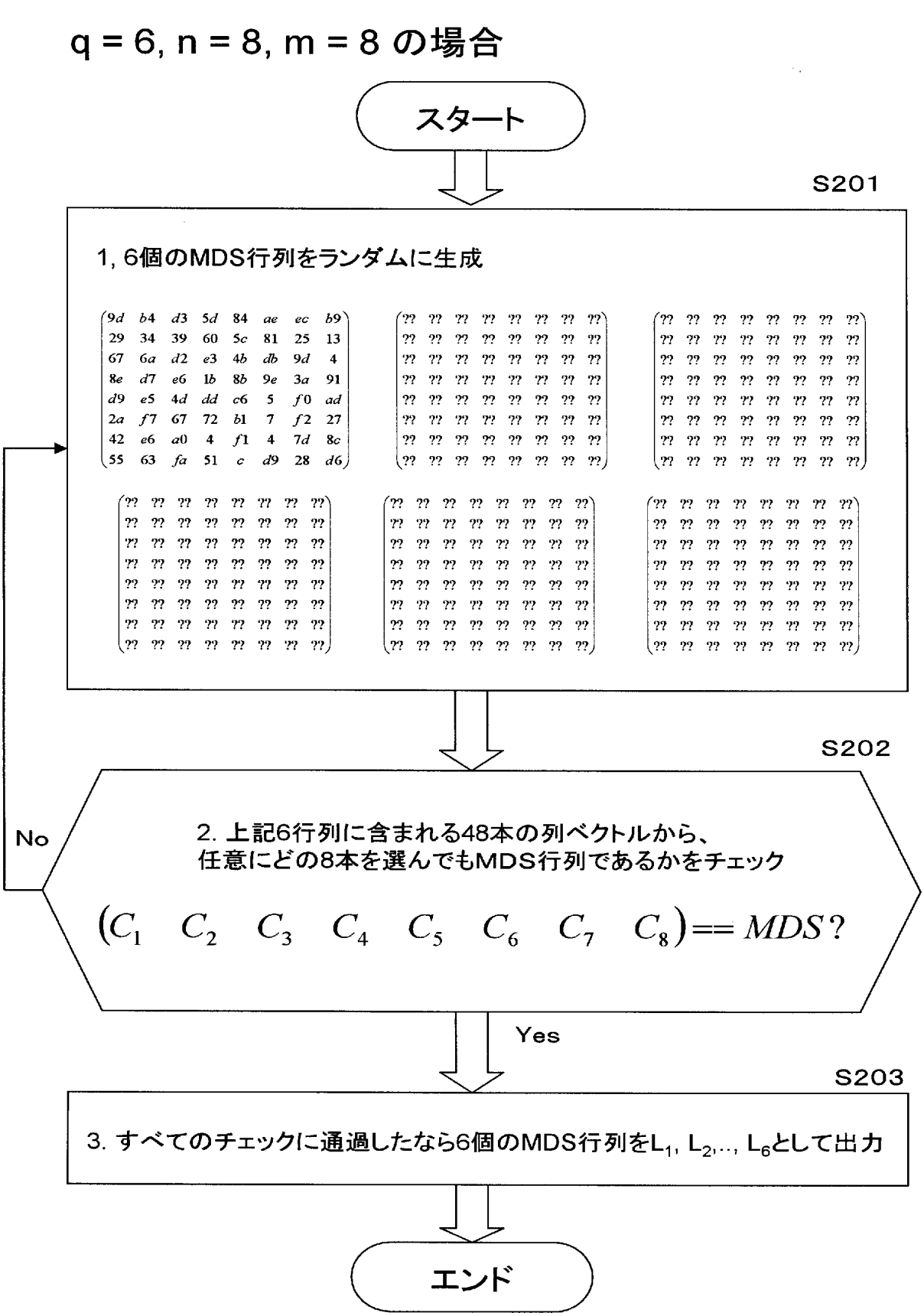
【図１０】


スタート
S21
q≦rのqを選択する
S22
q個のGF(2^n)上のm次MDS行列
L1, L2, ・・・, Lq を生成する
S23
2i－1 (i≦i≦r)段目の線形変換
行列MLT2iにL(i−1 mod q)+1を設定
S24
2i (i≦i≦r)段目の
線形変換行列MLT2iにMLT2r−2i+1を設定
エンド

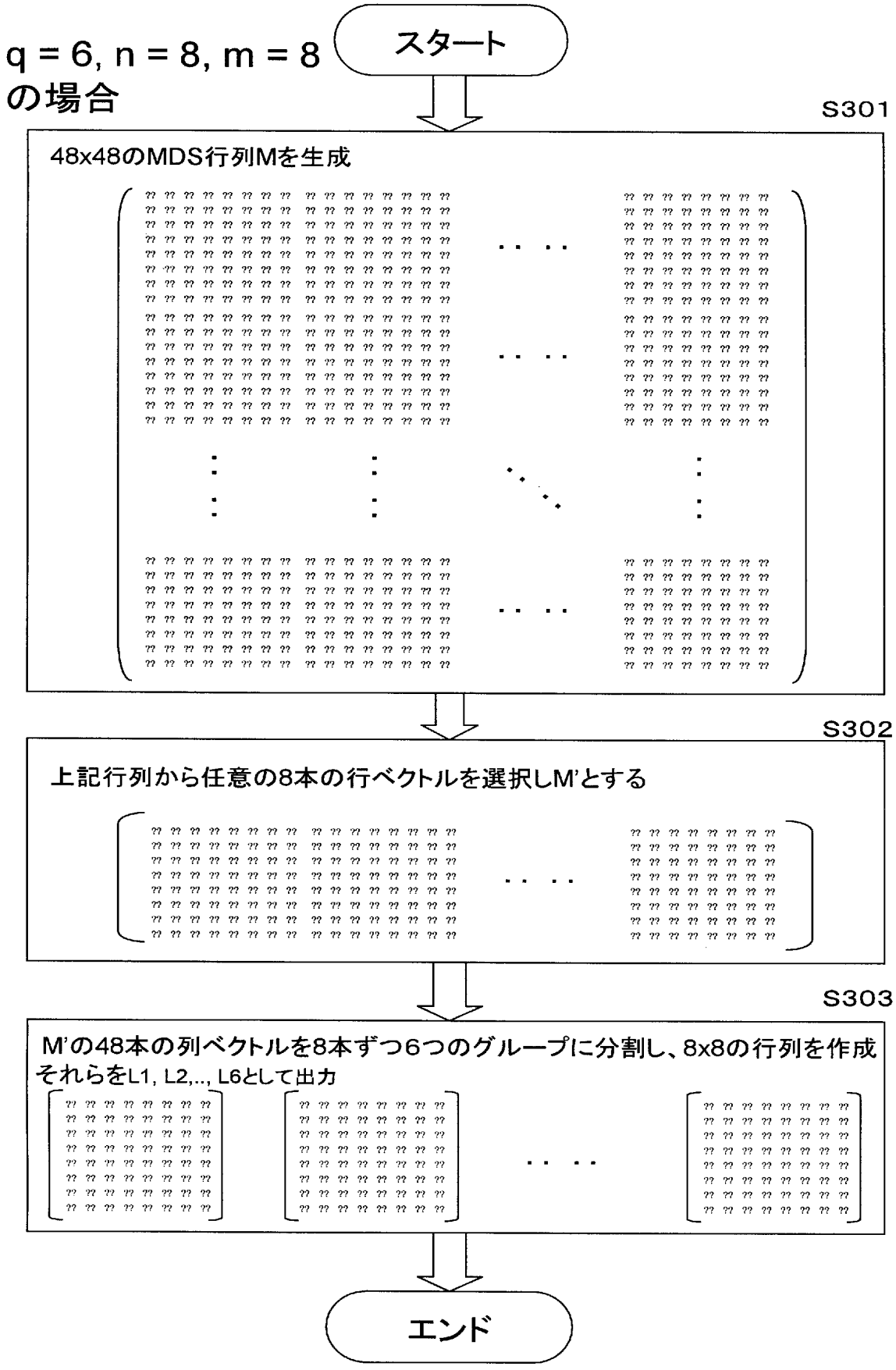
【図１１】


q = 6, n = 8, m = 8 の場合
スタート
S101
1, 6個のMDS行列をランダムに生成
S102
No
2. 上記6行列に含まれる48本の列ベクトルから、
任意にどの8本を選んでも線形独立であるかをチェック
(行列式のチェックアルゴリズムで可能、行列式が非ゼロで
あれば線形独立であると判別される)
$\det(C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \quad C_6 \quad C_7 \quad C_8) == 0?$
Yes
S103
3. チェックに通過したなら6個のMDS行列を$L_1, L_2, .., L_6$として出力
エンド

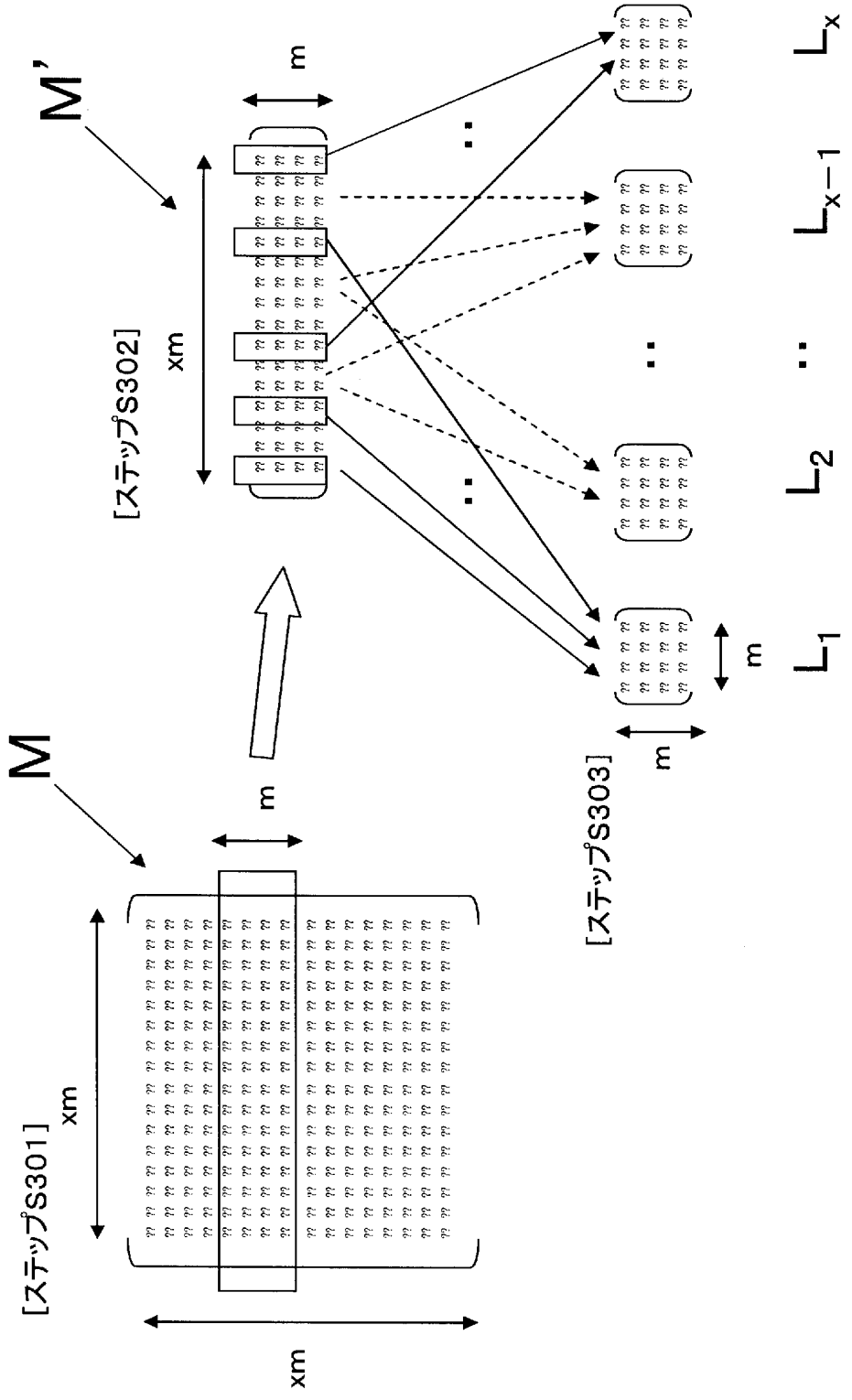
【図１２】


q = 6, n = 8, m = 8 の場合
スタート
S201
1, 6個のMDS行列をランダムに生成
S202
No
2. 上記6行列に含まれる48本の列ベクトルから、
任意にどの8本を選んでもMDS行列であるかをチェック
$(C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \quad C_6 \quad C_7 \quad C_8) == MDS?$
Yes
S203
3. すべてのチェックに通過したなら6個のMDS行列をL1, L2,.., L6として出力
エンド

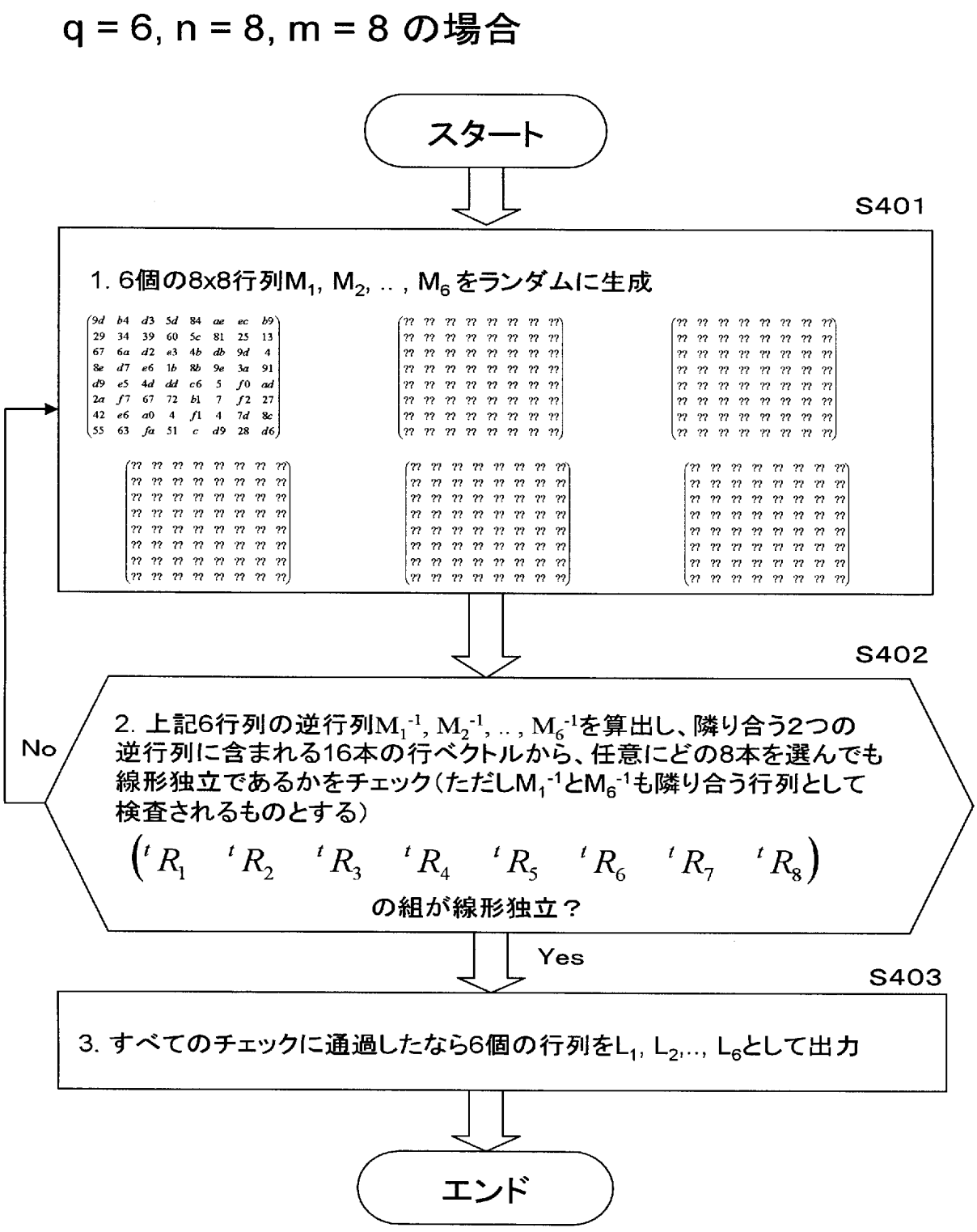
【図１３】


q = 6, n = 8, m = 8
の場合
スタート
S301
48x48のMDS行列Mを生成
S302
上記行列から任意の8本の行ベクトルを選択しM'とする
S303
M'の48本の列ベクトルを8本ずつ6つのグループに分割し、8x8の行列を作成
それらをL1, L2,.., L6として出力
エンド

【図１４】

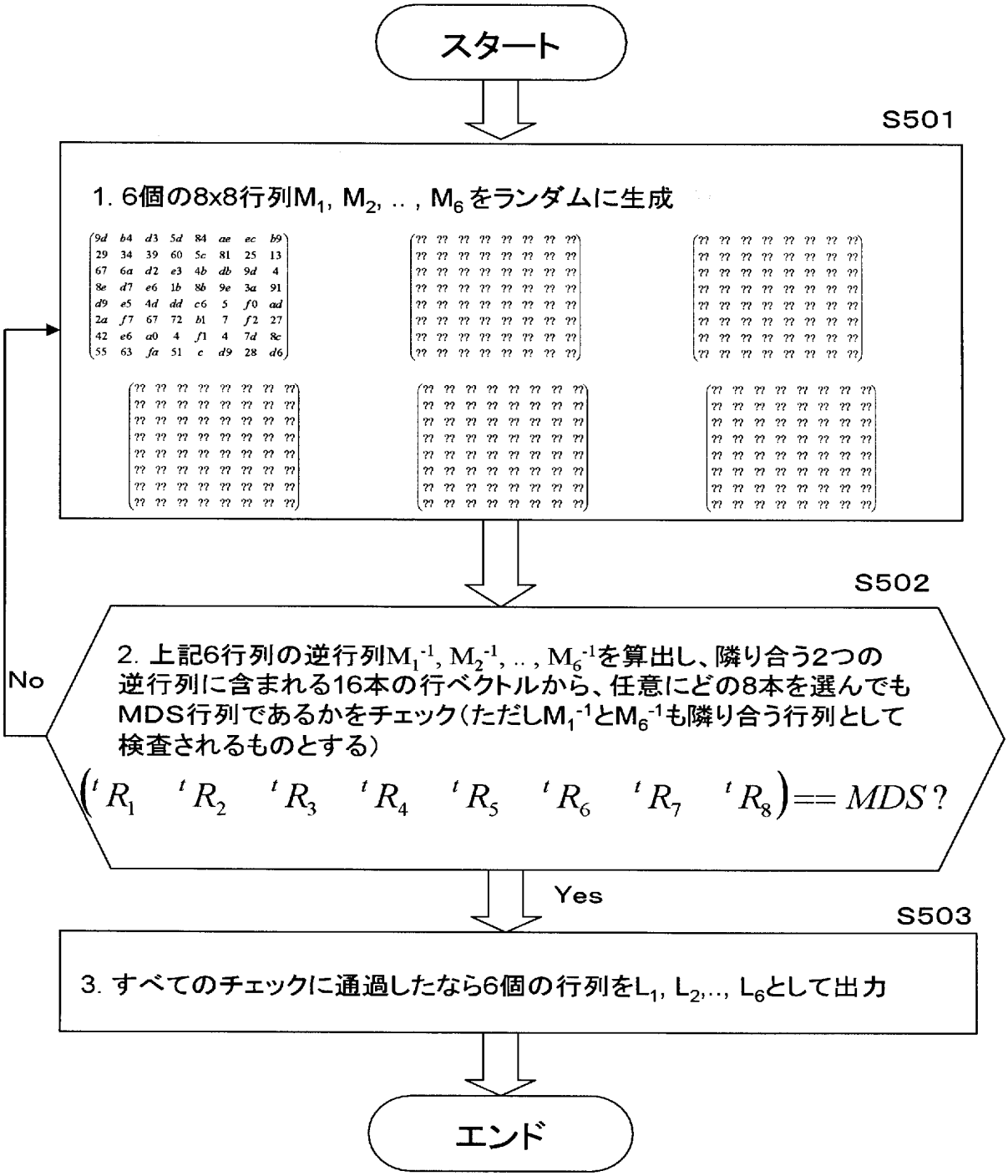


【図１５】

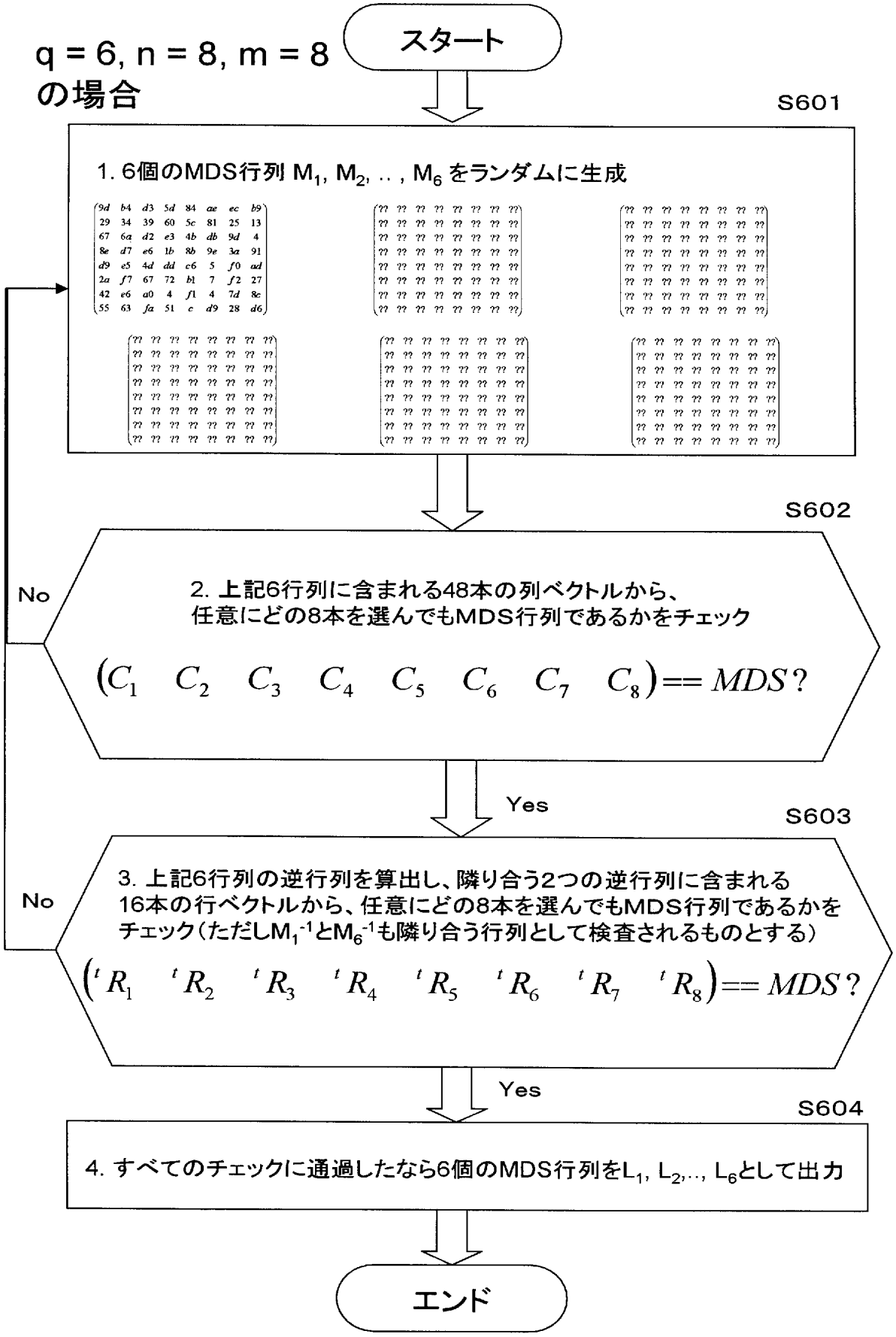

q = 6, n = 8, m = 8 の場合
スタート
S401
1. 6個の8x8行列M1, M2, .. , M6 をランダムに生成
S402
No
2. 上記6行列の逆行列M1-1, M2-1, .. , M6-1を算出し、隣り合う2つの逆行列に含まれる16本の行ベクトルから、任意にどの8本を選んでも線形独立であるかをチェック（ただしM1-1とM6-1も隣り合う行列として検査されるものとする）
の組が線形独立？
Yes
S403
3. すべてのチェックに通過したなら6個の行列をL1, L2,.., L6として出力
エンド

【図１６】

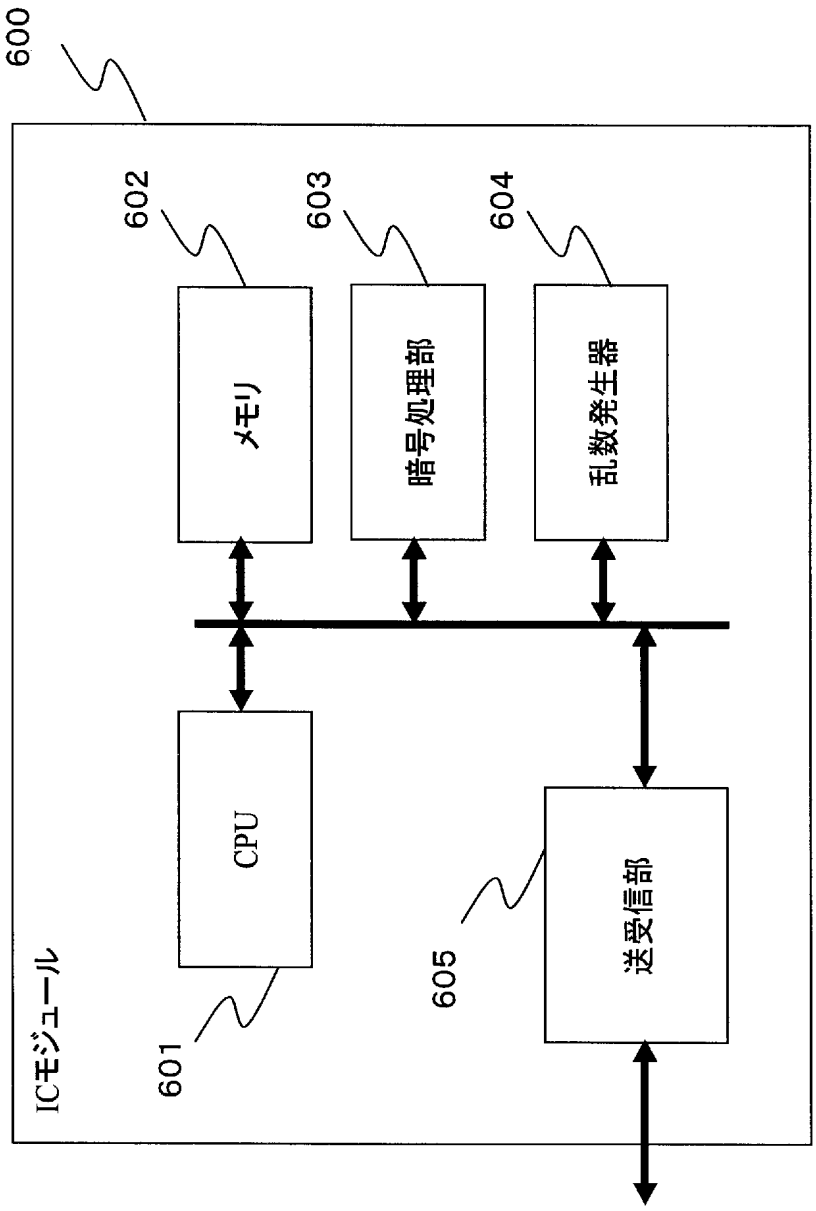
q = 6, n = 8, m = 8 の場合



【図１７】


q = 6, n = 8, m = 8
の場合
スタート
S601
1. 6個のMDS行列 M1, M2, .. , M6 をランダムに生成
S602
No
2. 上記6行列に含まれる48本の列ベクトルから、
任意にどの8本を選んでもMDS行列であるかをチェック
(C1 C2 C3 C4 C5 C6 C7 C8) == MDS?
Yes
S603
No
3. 上記6行列の逆行列を算出し、隣り合う2つの逆行列に含まれる
16本の行ベクトルから、任意にどの8本を選んでもMDS行列であるかを
チェック（ただしM1-1とM6-1も隣り合う行列として検査されるものとする）
(tR1 tR2 tR3 tR4 tR5 tR6 tR7 tR8) == MDS?
Yes
S604
4. すべてのチェックに通過したなら6個のMDS行列をL1, L2,.., L6として出力
エンド

【図１８】



【書類名】要約書
【要約】
【課題】　解析困難性を高めた、安全性の高い暗号処理装置および方法を実現する。
【解決手段】　非線形変換部および線形変換部を有するＳＰＮ型のＦ関数を、複数ラウンド繰り返し実行するＦｅｉｓｔｅｌ型共通鍵ブロック暗号処理において、複数ラウンド各々に対応するＦ関数の線形変換処理を、正方ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用した線形変換処理とする。少なくとも連続する偶数ラウンドおよび連続する奇数ラウンドの各々において設定される正方ＭＤＳ行列の逆行列に含まれる任意のｍ個の列ベクトルが正方ＭＤＳ行列である設定とする。本構成により、共通鍵ブロック暗号における線形攻撃に対する耐性の向上した暗号処理が実現される。
【選択図】　図１７

出願人履歴

０００００２１８５
19900830
新規登録
５９７０６２９９３

東京都品川区北品川６丁目７番３５号
ソニー株式会社